



---

Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Σερρών  
Σχολή Τεχνολογικών Εφαρμογών  
Τμήμα Πληροφορικής & Επικοινωνιών

---

**Πτυχιακή Εργασία:**

**Μανιόπουλος Σπυρίδων**

**Βούλγαρης Μιχαήλ**

**ΜΕΘΟΔΟΙ**

**Επιβλέπων:**

**Πάρις Μαστοροκώστας**

**Επ. Καθηγητής**

Σέρρες, Οκτώβριος 2005



Ανώτατο Τεχνολογικό Εκπαιδευτικό Ίδρυμα Σερρών

Σχολή Τεχνολογικών Εφαρμογών

Τμήμα Πληροφορικής & Επικοινωνιών

---

### ***Πτυχιακή Εργασία:***

Σπυρίδων Α. Μανιόπουλος (Α.Μ. 384)

Μιχαήλ Χ. Βούλγαρης (Α.Μ. 487)

**Θέμα:** *Ανάπτυξη και υλοποίηση εκπαιδευτικού υλικού  
για το γνωστικό αντικείμενο «Αριθμητική ανάλυση»*

### ***Επιβλέπων:***

Πάρις Μαστοροκώστας

*Επ. Καθηγητής*

Σέρρες, Οκτώβριος 2005

# Περιεχόμενα

## Κεφάλαιο 1

<b>Εισαγωγή &amp; βασική περιγραφή</b> .....	<b>4</b>
<b>1.1 Περιγραφή εργασίας</b> .....	<b>5</b>
<b>1.2 Αριθμητικές μέθοδοι</b> .....	<b>5</b>
1.2.1 Μέθοδος διχοτόμησης .....	6
1.2.2 Μέθοδος regula falsi & τέμνουσας .....	8
1.2.3 Μέθοδος του Newton .....	9
1.2.4 Μέθοδος Jacobi .....	12
1.2.5 Μέθοδος Gauss – Seidel .....	13
1.2.6 Μέθοδος ελαχίστων τετραγώνων .....	14

## Κεφάλαιο 2

<b>Λειτουργία και χρήση του προγράμματος</b> .....	<b>19</b>
<b>2.1 Εργαλεία προγραμματισμού</b> .....	<b>20</b>
<b>2.2 Περιγραφή προγράμματος</b> .....	<b>20</b>
2.2.1 Φόρμες .....	20
2.2.2 Γραφικές Παραστάσεις .....	52
2.2.3 Συναρτήσεις .....	53
<b>2.3 Χρήση του προγράμματος</b> .....	<b>53</b>

## Κεφάλαιο 3

<b>Παρουσίαση του κώδικα</b> .....	<b>54</b>
<b>3.1 Περιγραφή στοιχείων κώδικα</b> .....	<b>55</b>
<b>3.2 Κώδικας</b> .....	<b>59</b>

<b><u>Βιβλιογραφία</u></b> .....	<b>174</b>
----------------------------------	------------

# Κεφάλαιο 1ο

**ΕΙΣΑΓΩΓΗ & ΒΑΣΙΚΗ  
ΠΕΡΙΓΡΑΦΗ**

## 1.1 Περιγραφή εργασίας

Στην εργασία αυτή αναπτύχθηκε εκπαιδευτικό λογισμικό στη γλώσσα C++, το οποίο μπορεί να ενταχθεί στο εργαστήριο του μαθήματος «Αριθμητικές μέθοδοι σε προγραμματιστικό περιβάλλον». Το λογισμικό που παράχθηκε λειτουργεί σε παραθυρικό περιβάλλον και περιλαμβάνει:

**A)** Εφαρμογές των μεθόδων αριθμητικής ανάλυσης με συνοπτική παρουσίαση της θεωρίας (ως βοήθεια στην εφαρμογή),

**B)** Ασκήσεις αριθμητικών μεθόδων.

Το λογισμικό που παράχθηκε περιλαμβάνει σειρές ασκήσεων βασισμένες στις παρακάτω μεθόδους και από την εκτέλεση τους ο χρήστης μπορεί να κατανοήσει την χρήση τους.

## 1.2 Αριθμητικές μέθοδοι

Οι αριθμητικές μέθοδοι που χρησιμοποιούνται χωρίζονται σε τρεις κατηγορίες: **Επίλυσης εξισώσεων, εύρεσης ριζών και ελαχίστου τετραγώνου**. Αναλυτικότερα:

### Εύρεσης ριζών

- Μέθοδος διχοτόμησης
- Μέθοδος τέμνουσας
- Μέθοδος regula falsi
- Μέθοδος Newton

### Επίλυσης εξισώσεων

- Μέθοδος Jacobi
- Μέθοδος Gauss - Seidel

### Μέθοδος ελαχίστων τετράγωνων

### 1.2.1 Μέθοδος διχοτόμησης

#### ΟΡΙΣΜΟΣ ΟΡΙΩΝ ΚΑΙ ΔΙΧΟΤΟΜΗΣΗ

Σε μια ρίζα έχουν οριστεί τα όρια όταν στο διάστημα  $A, B$  οι συναρτήσεις έχουν αντίθετα πρόσημα. Αν η συνάρτηση είναι συνεχής, τότε τουλάχιστον μια ρίζα πρέπει να υπάρχει στο διάστημα  $A, B$ . Αν η συνάρτηση δεν είναι συνεχής, αλλά περιορισμένη, τότε αντί της ρίζας υπάρχει ένα σκαλοπάτι ασυνέχειας που περνάει από το 0. Για μαθηματικούς σκοπούς, αυτή μπορεί να είναι η ρίζα αφού η πορεία της είναι δυσδιάκριτη σε σχέση με την περίπτωση μιας συνεχούς συνάρτησης που περνάει από το μηδέν μεταξύ δύο γειτονικών αριθμών κινητής υποδιαστολής σε υψηλής ακρίβειας μηχανική αναπαράσταση. Μόνο για συναρτήσεις με ιδιομορφία υπάρχει η πιθανότητα μια ρίζα σε όρια να μην υπάρχει εκεί π.χ.:

$$f(x) = \frac{1}{x - c}$$

Κάποιοι αλγόριθμοι εύρεσης ρίζας μπορούν να συγκλίνουν στο  $c$  (στο παραπάνω παράδειγμα). Αν έχουμε μια συνάρτηση χωρίς στοιχεία, τότε δεν υπάρχει σίγουρος τρόπος οριοθέτησης, ή ακόμα και υπολογισμού της ρίζας. Για παράδειγμα: η εύρεση των δύο πραγματικών ριζών μιας συνάρτησης, που πέφτει κάτω από το 0 μόνο στο πολύ μικρό διάστημα:

$$x = \pi \pm 10^{-667}$$

Άλλο πρόβλημα είναι η οριοθέτηση του ελάχιστου της συνάρτησης. Υπάρχει η πιθανότητα να δώσουμε μία μέθοδο που πάντα πετυχαίνει, για παράδειγμα «Πηγαίνετε κατηφορικά, χρησιμοποιώντας όλο και πιο μεγάλα σκαλιά μέχρι η συνάρτηση να αλλάξει φορά (ανηφορική)». Δεν υπάρχουν ανάλογες μέθοδοι για την εύρεση ρίζας. Η παραπάνω μέθοδος μπορεί να ανατραπεί χρησιμοποιώντας μια απλή συνάρτηση.

## ΔΙΧΟΤΟΜΗΣΗ

Αν ξέρουμε ότι ένα διάστημα περιέχει μία ρίζα, τότε υπάρχουν διάφορες μέθοδοι για να τη βρούμε. Αυτές οι μέθοδοι προσφέρουν διαφορετικές ταχύτητες και ακρίβειες για τον υπολογισμό της ρίζας. Δυστυχώς μέθοδοι που συγκλίνουν αργά μέχρι πολύ αργά, αντιθέτως από τις μεθόδους που συγκλίνουν γρήγορα, μπορούν στην καλύτερη περίπτωση να εκσφενδονιστούν στο άπειρο χωρίς προειδοποίηση, αν δεν ληφθούν μέτρα.

Η μέθοδος της διχοτόμησης δεν μπορεί να αποτύχει. Γι' αυτό και χρησιμοποιείται για προβλήματα «κακής συμπεριφοράς» της συνάρτησης. Η ιδέα είναι απλή. Είναι γνωστό ότι σε συγκεκριμένο διάστημα και σημείο η συνάρτηση περνάει από το μηδέν επειδή αλλάζει πρόσημο. Εκτιμείστε την συνάρτηση στο μέσο του διαστήματος και ελέγξτε πρόσημο της. Χρησιμοποιείστε το μεσοδιάστημα για να αντικαταστήσετε το όριο με το ίδιο πρόσημο. Έπειτα από κάθε επανάληψη τα όρια που περιέχουν τη ρίζα μειώνονται κατά συντελεστή του δύο. Αν μετά από  $n$  επαναλήψεις η ρίζα βρίσκεται μέσα σε διάστημα μεγέθους  $e_n$ , τότε μετά την επανάληψη θα βρίσκεται σε ένα διάστημα μεγέθους

$$e_{n+1} = \frac{e_n}{2}$$

ακριβώς. Έτσι γνωρίζουμε από πριν τον αριθμό των επαναλήψεων που χρειάζεται για να επιτύχουμε την δεδομένη ακρίβεια στη λύση,

$$n = \log_2 \frac{e_0}{e}$$

όπου  $e_0$  το μέγεθος του αρχικού διαστήματος οριοθέτησης και  $E$  η επιθυμητή ακρίβεια. Η διχοτόμηση πρέπει να επιτύχει. Αν στο διάστημα υπάρχουν δύο ή περισσότερες ρίζες, τότε η διχοτόμηση θα βρει τουλάχιστον μία. Αν στο διάστημα δεν υπάρχει καμία ρίζα τότε θα συγκλίνει στη μοναδικότητα. Όταν η μέθοδος συγκλίνει σαν ένας συντελεστής (μικρότερος από 1) και χρονίζει την προηγούμενη αβεβαιότητα στην πρώτη δύναμη (όπως στην διχοτόμηση), λέγεται ότι συγκλίνει γραμμικά.

Σε άλλες συναφείς «γραμμικές» μεθόδους η σύγκλιση ορίζεται ως «εκθετική» ή «γεωμετρική». Αυτό δεν είναι κακό γιατί γραμμική σύγκλιση σημαίνει ότι διαδοχικοί σημαντικοί ψηφιοχαρακτήρες κερδίζουν γραμμικά με υπολογιστική προσπάθεια.

Ακολουθώς θα μιλήσουμε για τα κριτήρια σύγκλισης. Είναι σημαντικό να θυμόμαστε ότι οι υπολογιστές χρησιμοποιούν δεκαδικά ψηφία για να αναπαραστήσουν τους δεκαδικούς αριθμούς. Μπορεί η μέθοδος να περνά αναλυτικά από το 0, αλλά η τιμή στον υπολογιστή να μην πάρει τιμή 0, για κάθε δεκαδικό όρισμα. Κάποιος πρέπει να αποφασίσει ποια ακρίβεια στη ρίζα είναι εφικτή. Σύγκλιση με ακρίβεια  $10^{-6}$  είναι απόλυτα λογική για ρίζα κοντά στο 1, αλλά μη εφικτή αν η ρίζα βρίσκεται κοντά στο  $10^{26}$ . Πρέπει να σκεφτούμε πώς να ορίσουμε σύγκλιση με συγγενικό (κλασματικό) κριτήριο, αλλά αυτό δεν δουλεύει για ρίζες κοντά στο 0.

### **1.2.2 Μέθοδος regula falsi & τέμνουσας**

Για συναρτήσεις που είναι σταθερές κοντά στη ρίζα, οι μέθοδοι τέμνουσας και regula falsi συγκλίνουν γενικά γρηγορότερα από την διχοτόμηση. Και στις δύο μεθόδους η συνάρτηση θεωρείται προσεγγιστικά γραμμική, και η επόμενη βελτίωση (προσέγγιση) της ρίζας είναι στο σημείο που η προσεγγιστική γραμμή περνά την γραμμή των αξόνων. Μετά από κάθε επανάληψη ένα από τα όρια αλλάζει με την τελευταία εκτίμηση της ρίζας.

Η ρίζα εκτιμάται από τον τύπο:

$$x_i = b - \frac{b-a}{f(b)-f(a)} f(b)$$

Η μόνη διαφορά μεταξύ των μεθόδων τέμνουσας και regula falsi, είναι ότι η τέμνουσα συγκρατεί την πιο πρόσφατη από τις προηγούμενες εκτιμήσεις, ενώ η regula falsi συγκρατεί την προηγούμενη εκτίμηση για την οποία η συνάρτηση έχει αντίθετο πρόσημο από την συνάρτηση της καλύτερης τρέχουσας ρίζας, έτσι ώστε τα δύο σημεία να συνεχίζουν οριοθετούν την ρίζα. Μαθηματικά, η μέθοδος τέμνουσας συγκλίνει γρηγορότερα κοντά σε ρίζα μίας επαρκούς συνεχής συνάρτησης. Ο ρυθμός σύγκλισης μπορεί να φτάσει ως τη «χρυσή αναλογία» 1,618... έτσι ώστε :



$$\lim_{k \rightarrow \infty} |e_k + 1| \approx \text{const} \times |e_k|^{1.618}$$

Η τέμνουσα ωστόσο έχει το μειονέκτημα ότι η ρίζα δεν παραμένει μέσα στα όρια. Για συνάρτηση που δεν είναι επαρκώς συνεχής, ο αλγόριθμος μπορεί να μην συγκλίνει. Η τοπική συμπεριφορά της μεθόδου μπορεί να τη στείλει στο άπειρο.

Η regula falsi, παρόλο που μερικές φορές κρατά παλαιότερη παρά νεότερη εκτίμηση, έχει χαμηλότερο ρυθμό σύγκλισης.

### **1.2.3 Μέθοδος του Newton**

Πιθανώς η πιο φημισμένη από όλες τις μονοδιάστατες μεθόδους εύρεσης ρίζας είναι η μέθοδος Newton ή αλλιώς Newton – Raphson. Η μέθοδος Newton καλύπτεται από τις άλλες μεθόδους, γιατί απαιτεί εκτίμηση της συνάρτησης και της παραγώγου της, στα κυρίως σημεία  $x$ . Η μέθοδος Newton δημιουργείται γεωμετρικά επιμηκύνοντας την εφαπτόμενη στο τρέχων σημείο  $X_i$  μέχρι να περάσει το 0, τότε ορίζουμε την επόμενη εκτίμηση της ρίζας  $X_{i+1}$  στη τετμημένη του περάσματος από το 0. Αλγοριθμικά, η μέθοδος προέρχεται από την ανάπτυξη των σειρών Taylor για μια συνάρτηση που βρίσκεται στη «γειτονιά» ενός σημείου,

$$f(x + \delta) \approx f(x) + f'(x)\delta + \frac{f''(x)}{2}\delta^2 + \dots$$

για αρκετά μικρές τιμές του  $\delta$ , και για ομαλές συναρτήσεις, οι όροι πέρα από την γραμμική είναι ασήμαντοι, όποτε το  $f(x+\delta)=0$  οδηγεί στο:

$$\delta = -\frac{f(x)}{f'(x)}$$

Η μέθοδος Newton δεν είναι περιορισμένη μόνο σε μία διάσταση. Εύκολα γενικεύεται σε περισσότερες διαστάσεις.

Αρκετά μακριά από μια ρίζα, όπου οι υψηλής τάξης όροι είναι σημαντικοί, η μέθοδος Newton μπορεί να δώσει στιλπνές, ανακριβείς και ασήμαντες διορθώσεις. Για παράδειγμα, η πρώτη εκτίμηση της ρίζας μπορεί να είναι τόσο μακριά από την πραγματική ρίζα σαν να αφήναμε το διάστημα έρευνας να περιέχει το τοπικό μέγιστο

και ελάχιστο της συνάρτησης. Αυτό μπορεί να είναι το τέλος της μεθόδου. Αν η επανάληψη εκτιμήσει την ρίζα κοντά σε ένα τέτοιο τοπικό έσχατο, ώστε η πρώτη παράγωγος να έχει εξαφανιστεί, τότε η λύση της μεθόδου μπορεί να ξεφεύγει τόσο πολύ ώστε να μην υπάρχει ελπίδα ανάκαμψης. Η μέθοδος Newton μπορεί να είναι καταστροφική αν χρησιμοποιηθεί σε λάθος περιπτώσεις.

Γιατί η μέθοδος Newton θεωρείται πανίσχυρη; Η απάντηση βρίσκεται στο ρυθμό σύγκλισης της μεθόδου. Μέσα σε ένα μικρό διάστημα  $\varepsilon$  του  $x$  η συνάρτηση και η παραγωγός της είναι περίπου:

$$f(x + e) = f(x) + ef'(x) + e^2 \frac{f''(x)}{2} + \dots,$$

$$f'(x + e) = f'(x) + ef''(x) + \dots$$

Από την μέθοδο Newton:

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)},$$

έτσι ώστε:

$$e_{i+1} = e_i - \frac{f(x_i)}{f'(x_i)}$$

Όταν μια τυχαία λύση  $X_i$  διαφέρει από την πραγματική κατά  $E_i$ , μπορούμε να χρησιμοποιήσουμε την πρώτη εξίσωση για να εκφράσουμε τις  $f(X_i)$ ,  $f'(X_i)$  στην δεύτερη εξίσωση στα πλαίσια του  $E_i$  και τις παραγώγους στην ίδια ρίζα. Το αποτέλεσμα είναι μια περιοδική σχέση για τις παρεκκλίσεις των λύσεων:

$$e_{i+1} = -e_i \frac{f''(x)}{2f'(x)}.$$

Η παραπάνω εξίσωση δείχνει ότι η μέθοδος Newton συγκλίνει τετραγωνικά. Κοντά σε μια ρίζα, ο αριθμός των σημαντικών ψηφίων διπλασιάζεται σε κάθε βήμα. Αυτή η ισχυρή ιδιότητα σύγκλισης κάνει την μέθοδο Newton πρώτη επιλογή για συναρτήσεις των οποίων οι παράγωγοι έχουν μπορούν να εκτιμηθούν αποδοτικά και είναι συνεχείς και μη μηδενικοί στη γειτονία μίας ρίζας. Ακόμα και εκεί που η Newton αποφεύγεται λόγω των πρώτων σταδίων της σύγκλισης, είναι συνηθισμένο

να «γυαλίζουμε» την ρίζα εκτελώντας ένα ή δυο επαναλήψεις της Newton, που μπορούν να πολλαπλασιαστούν με 2 ή 4 στο νούμερο των σημαντικών ψηφιοχαρακτήρων.

Για μια αποδοτική αντίληψη της μεθόδου, ο χρήστης δίνει μία ρουτίνα που υπολογίζει την συνάρτηση και την παράγωγο της στο σημείο  $x$ . Η μέθοδος Newton μπορεί να εφαρμοστεί χρησιμοποιώντας μια αριθμητική διαφορά για να προσεγγίσει την πραγματική τοπική παράγωγο,

$$f'(x) \approx \frac{f(x+dx) - f(x)}{dx}$$

Αυτή όμως δεν είναι η προτεινόμενη διαδικασία για τους ακόλουθους λόγους:

- α) Εκτελούνται δύο εκτιμήσεις συναρτήσεων, έτσι στην καλύτερη των περιπτώσεων ο ρυθμός σύγκλισης είναι ρίζα 2,
- β) Αν διαλέξουμε πολύ μικρή τιμή του  $dx$  θα καταλήξουμε σε αδιέξοδο, ενώ αν επιλέξουμε μεγάλη τιμή τότε ο ρυθμός σύγκλισης θα είναι γραμμικός, δηλαδή όχι καλύτερα από τη αρχική εκτίμηση του  $f'(x)$  για όλες τις επόμενες επαναλήψεις. Γι' αυτό η μέθοδος Newton με παραγώγους, υπερνικάται από την μέθοδο της τέμνουσας (Σε περιπτώσεις όμως πολλών διαστάσεων η Newton λαμβάνεται σοβαρότερα υπόψη).

Αφού οι συνολικές ιδιότητες σύγκλισης της μεθόδου Newton είναι φτωχές, είναι προτιμότερο να σχεδιάσουμε μία ασφαλή ρουτίνα που χρησιμοποιεί ένα συνδυασμό διχοτόμησης και Newton. Ο υβριδικός αλγόριθμος εκτελεί μια επανάληψη διχοτόμησης κάθε φορά που η Newton λαμβάνει μία ρίζα έξω από τα όρια, ή όταν η Newton δεν μειώνει αρκετά γρήγορα το μέγεθος των ορίων.

Για πολλές συναρτήσεις η παράγωγος συγκλίνει με την απαιτούμενη ακρίβεια γρηγορότερα από ότι η ίδια η συνάρτηση. Όταν αυτό συμβεί, κάποιος πρέπει σύντομα να ενημερώσει την παράγωγο. Αυτό το μονοπάτι ακολουθείται μόνο όταν έχει γίνει κατανοητή η συμπεριφορά της συνάρτησης, αλλά αυξάνει την ταχύτητα των πράξεων όταν ο υπολογισμός της παράγωγου είναι επίπονος (επίσημα, αυτό κάνει την σύγκλιση μόνο γραμμική, αλλά αν δεν αλλάζει η παράγωγος, δεν μπορεί να γίνει τίποτε άλλο).

### 1.2.4 Μέθοδος Jacobi

Θεωρούμε γραμμικό σύστημα  $n \times n$ :

$$A_{11}X_1 + A_{12}X_2 + \dots + A_{1n}X_n = B_1$$

$$A_{21}X_1 + A_{22}X_2 + \dots + A_{2n}X_n = B_2$$

.....

$$A_{n1}X_1 + A_{n2}X_2 + \dots + A_{nn}X_n = B_n$$

όπου οι συντελεστές  $A_{ij}$ ,  $1 \leq i, j \leq n$ , και τα δευτέρα μέλη  $B_1, \dots, B_n$  είναι δεδομένα, πραγματικοί αριθμοί. Το ζητούμενο είναι μία  $n$ -αδα αριθμών να ικανοποιούν το σύστημα, δηλαδή μία λύση του. Ορίζουμε το σύστημα ως πίνακες  $A[n, n]$  και τα διανύσματα  $B[n]$  και  $X[n]$  και γράφουμε το γραμμικό σύστημα στη μορφή:  $A \cdot X = B$ . Σκοπός των μεθόδων είναι η εύρεση των νέων  $X$  τιμών (λύσεις) ορίζοντας αρχικές τιμές στον πίνακα  $X$  και η επανάληψη αυτής της διαδικασίας.

Η εύρεση των νέων τιμών γίνεται με την βοήθεια των παρακάτω αλγεβρικών πράξεων:

$$\begin{aligned}
 X_1^{(new)} &= 0 - \frac{A_{12}}{A_{11}} X_2^{(old)} - \frac{A_{13}}{A_{11}} X_3^{(old)} - \dots - \frac{A_{1n}}{A_{11}} X_n^{(old)} + \frac{B_1}{A_{11}} \\
 &\quad \vdots \\
 X_j^{(new)} &= \frac{A_{j1}}{A_{jj}} X_1^{(old)} - \frac{A_{j2}}{A_{jj}} X_2^{(old)} - \dots - \frac{A_{j(j-1)}}{A_{jj}} X_{j-1}^{(old)} - 0 - \frac{A_{j(j+1)}}{A_{jj}} X_{j+1}^{(old)} - \frac{A_{jn}}{A_{jj}} X_n^{(old)} + \frac{B_j}{A_{jj}} \\
 &\quad \vdots \\
 X_n^{(new)} &= -\frac{A_{n1}}{A_{nn}} X_1^{(old)} - \dots - \frac{A_{n(n-1)}}{A_{nn}} X_{n-1}^{(old)} - 0 + \frac{B_n}{A_{nn}} .
 \end{aligned}$$

Στη μέθοδο Jacobi οι τιμές  $X$  είναι οι αρχικές τιμές για την πρώτη επανάληψη και οι τιμές που προέκυψαν από την προηγούμενη επανάληψη για τις επόμενες επαναλήψεις.

Για να ορίσουμε την ακρίβεια, καθώς και τις επαναλήψεις της μεθόδου, με την οποία βρίσκουμε τις λύσεις, χρησιμοποιούμε τον τύπο της ευκλείδειας νόρμας για την παλιά και νέα λύση,

$$\sqrt{\sum_{j=1}^n (X_j^{(new)} - X_j^{(old)})^2} < \rho$$

όπου  $\rho$  το όριο της διαφοράς.

### 1.2.5 Μέθοδος Gauss - Seidel

Θεωρούμε γραμμικό σύστημα  $n \times n$ :

$$A_{11}X_1 + A_{12}X_2 + \dots + A_{1n}X_n = B_1$$

$$A_{21}X_1 + A_{22}X_2 + \dots + A_{2n}X_n = B_2$$

.....

$$A_{n1}X_1 + A_{n2}X_2 + \dots + A_{nn}X_n = B_n$$

όπου οι συντελεστές  $A_{ij}$ ,  $1 \leq i, j \leq n$ , και τα δευτέρα μέλη  $B_1, \dots, B_n$  είναι δεδομένα, πραγματικοί αριθμοί. Το ζητούμενο είναι μία  $n$ -αδα αριθμών να ικανοποιούν το σύστημα, δηλαδή μία λύση του. Ορίζουμε το σύστημα ως πίνακες  $A[n, n]$  και τα διανύσματα  $B[n]$  και  $X[n]$  και γράφουμε το γραμμικό σύστημα στη μορφή:  $A \cdot X = B$ . Σκοπός των μεθόδων είναι η εύρεση των νέων  $X$  τιμών (λύσεις) ορίζοντας αρχικές τιμές στον πίνακα  $X$  και η επανάληψη αυτής της διαδικασίας.

Η εύρεση των νέων τιμών γίνεται με την βοήθεια των παρακάτω αλγεβρικών πράξεων:

$$X_1^{(new)} = 0 - \frac{A_{12}}{A_{11}} X_2^{(old)} - \frac{A_{13}}{A_{11}} X_3^{(old)} - \dots - \frac{A_{1n}}{A_{11}} X_n^{(old)} + \frac{B_1}{A_{11}}$$

$$\vdots$$

$$X_j^{(new)} = \frac{A_{j1}}{A_{jj}} X_1^{(new)} - \frac{A_{j2}}{A_{jj}} X_2^{(new)} - \dots - \frac{A_{j(j-1)}}{A_{jj}} X_{j-1}^{(new)} - 0 - \frac{A_{j(j+1)}}{A_{jj}} X_{j+1}^{(new)} - \frac{A_{jn}}{A_{jj}} X_n^{(new)} + \frac{B_j}{A_{jj}}$$

$$\vdots$$

$$X_n^{(new)} = -\frac{A_{n1}}{A_{nn}} X_1^{(new)} - \dots - \frac{A_{n(n-1)}}{A_{nn}} X_{n-1}^{(new)} - 0 + \frac{B_n}{A_{nn}}$$

Στη μέθοδο Gauss Seidel οι τιμές  $X$  είναι οι αρχικές τιμές για την πρώτη ακολουθία πράξεων και οι τιμές που προέκυψαν για τις επόμενες ακολουθίες. Στις επόμενες επαναλήψεις χρησιμοποιούνται με τον ίδιο τρόπο οι τιμές  $X$  που προέκυψαν.

Για να ορίσουμε την ακρίβεια, καθώς και τις επαναλήψεις της μεθόδου, με την οποία βρίσκουμε τις λύσεις, χρησιμοποιούμε τον τύπο της ευκλείδειας νόρμας για την παλιά και νέα,

$$\sqrt{\sum_{j=1}^n (X_j^{(new)} - X_j^{(old)})^2} < \rho$$

όπου  $\rho$  το όριο της διαφοράς.

### **1.2.6 Μέθοδος ελαχίστων τετραγώνων**

Θεωρούμε το πρόβλημα της εφαρμογής ενός συνόλου  $N$  σημείων  $(X, Y)$  στο μοντέλο της ευθείας:

$$y(x) = y(x; a, b) = a + bx.$$

Το πρόβλημα αυτό ονομάζεται «γραμμική παλινδρόμηση», μια ορολογία που προέρχεται από τα πρώτα χρόνια των επιστημών. Υποθέτουμε ότι η αβεβαιότητα  $\sigma$  σε σχέση με την μέτρηση  $Y$  είναι γνωστή, και ότι τα  $X$  (τιμές της εξαρτώμενης μεταβλητής) είναι επίσης είναι γνωστά.

Για να υπολογίσουμε πόσο καλά το μοντέλο συμφωνεί με τα δεδομένα, χρησιμοποιούμε την «χ-τετράγωνο» ποσοτική συνάρτηση, που σε αυτή την περίπτωση είναι:

$$x^2(a, b) = \sum_{i=0}^{N-1} \left( \frac{y_i - a - bx_i}{\sigma_i} \right)^2$$

Αν τα υπολογιστικά λάθη είναι κανονικά κατανομημένα, τότε αυτή η ποσοτική συνάρτηση μπορεί να δώσει την μέγιστη πιθανότητα κρίσης των παραμέτρων  $a$  και  $b$ . Αν τα λάθη δεν είναι κανονικά κατανομημένα, τότε οι

παράμετροι δεν έχουν υπολογιστεί με την μέγιστη πιθανότητα, αλλά μπορεί η συνάρτηση να είναι χρήσιμη με την πρακτική έννοια.

Η παραπάνω συνάρτηση μειώνεται για να υπολογίσουμε τα  $a$  και  $b$ . Στο ελάχιστο, η παραγωγός της  $\chi$  προστατεύοντας τα  $a$  και  $b$  εξαφανίζεται.

$$0 = \frac{\partial \chi^2}{\partial a} = -2 \sum_{i=0}^{N-1} \frac{y_i - a - bx_i}{\sigma_i^2}$$

$$0 = \frac{\partial \chi^2}{\partial b} = -2 \sum_{i=0}^{N-1} \frac{x_i (y_i - a - bx_i)}{\sigma_i^2}$$

Αυτοί οι όροι μπορούν να ξαναγραφούν σε μια πιο απλή μορφή, αν ορίσουμε τα παραπάνω σύνολα:

$$S = \sum_{i=0}^{N-1} \frac{1}{\sigma_i^2} \quad S_x = \sum_{i=0}^{N-1} \frac{x_i}{\sigma_i^2} \quad S_y = \sum_{i=0}^{N-1} \frac{y_i}{\sigma_i^2}$$

$$S_{xx} = \sum_{i=0}^{N-1} \frac{x_i^2}{\sigma_i^2} \quad S_{xy} = \sum_{i=0}^{N-1} \frac{x_i y_i}{\sigma_i^2}$$

Με αυτούς τους ορισμούς έχουμε:

$$aS + bS_x = S_y$$

$$aS_x + bS_{xx} = S_{xy}$$

Οι λύσεις αυτών των δυο εξισώσεων υπολογίζεται σαν

$$\Delta \equiv SS_{xx} - (S_x)^2$$

$$a = \frac{S_{xx}S_y - S_xS_{xy}}{\Delta}$$

$$b = \frac{SS_{xy} - S_xS_y}{\Delta}$$

Η παραπάνω εξίσωση δίνει την καλύτερη λύση για τις παραμέτρους  $a$  και  $b$ .

Ωστόσο δεν τελειώσαμε ακόμη. Πρέπει να υπολογίσουμε τις αβεβαιότητες στους υπολογισμούς των  $a$  και  $b$ , αφού σίγουρα τα υπολογιστικά λάθη, πρέπει να δημιουργούν κάποια αβεβαιότητα στον καθορισμό αυτών των παραμέτρων. Αν τα

δεδομένα είναι ανεξάρτητα, τότε το κάθε σημείο εισάγει το δικό του κομμάτι αβεβαιότητας στις παραμέτρους. Λαμβάνοντας υπόψη την εξάπλωση των λαθών φαίνεται ότι η διακύμανση  $\sigma_f^2$  σε κάθε τιμή της συνάρτησης είναι:

$$\sigma_f^2 = \sum_{i=0}^{N-1} \sigma_i^2 \left( \frac{\partial f}{\partial y_i} \right)^2$$

Για την ευθεία, οι παράγωγοι των  $a$  και  $b$  λαμβάνοντας υπόψη το  $Y_i$ , μπορούν να υπολογιστούν κατευθείαν από τη λύση:

$$\frac{\partial a}{\partial y_i} = \frac{S_{xx} - S_x X_i}{\sigma_i^2 \Delta}$$

$$\frac{\partial b}{\partial y_i} = \frac{S x_i - S_x}{\sigma_i^2 \Delta}$$

Αθροίζοντας όλα τα σημεία, παίρνουμε:

$$\sigma_a^2 = \frac{S_{xx}}{\Delta}$$

$$\sigma_b^2 = \frac{S}{\Delta}$$

που είναι οι διακυμάνσεις των  $a$  και  $b$ , αντίστοιχα. Επίσης ένα επιπλέον νούμερο χρειάζεται για να υπολογίσουμε κατάλληλα την πιθανή αβεβαιότητα των τιμών των παραμέτρων. Το νούμερο είναι ο συντελεστής διακύμανσης των  $a$  και  $b$  και δίνεται από τον τύπο:

$$Cov(a, b) = -\frac{S_x}{\Delta}$$

Ο συντελεστής συσχέτισης μεταξύ της αβεβαιότητας στο  $a$  και το  $b$ , που είναι ένα νούμερο μεταξύ του  $-1$  και  $1$ , ακολουθείται από το:

$$r_{ab} = \frac{-S_x}{\sqrt{SS_{xx}}}$$

Μια θετική τιμή του  $r_{ab}$  δείχνει ότι τα λάθη στο  $a$  και το  $b$  συνήθως έχουν το ίδιο πρόσημο, ενώ μία αρνητική τιμή δείχνει ότι τα λάθη είναι ασυσχέτιστα, και συνήθως έχουν αντίθετα πρόσημα.



Ακόμα, όμως δεν τελειώσαμε. Πρέπει να υπολογίσουμε πόσο καλά τα δεδομένα ταιριάζουν στο μοντέλο. Χωρίς αυτόν τον υπολογισμό, δεν έχουμε την παραμικρή ένδειξη ότι οι παράμετροι  $\alpha$  και  $\beta$  έχουν κάποια σημασία. Η πιθανότητα  $Q$  η τιμή  $\chi$  να συγκλίνει κατά λάθος είναι:

$$Q = \text{gammap}\left(\frac{N-2}{2}, \frac{x^2}{2}\right)$$

Εδώ η  $\text{gammap}$  είναι η ρουτίνα μας για την ελλιπή γάμα συνάρτηση  $Q(\alpha, x)$ . Αν η  $Q$  είναι μεγαλύτερη, για παράδειγμα από το 0.1, τότε τα δεδομένα ταιριάζουν στο μοντέλο. Αν είναι μεγαλύτερη π.χ. από 0.001, τότε τα δεδομένα ταιριάζουν αν τα λάθη δεν είναι κανονικοποιημένα ή αν έχουν υποεκτιμηθεί. Αν το  $Q$  είναι μικρότερο του 0.001, τότε το μοντέλο ή η διαδικασία υπολογισμού δεν είναι σίγουρα. Αν δεν γνωρίζετε τα ατομικά υπολογιστικά λάθη των σημείων  $y_i$ , και συνεχίζετε να χρησιμοποιείτε την εξίσωση:

$$\sigma^2 = \sum_{i=0}^{N-1} \frac{[y_i - y(x_i)]^2}{(N-M)}$$

για να υπολογίζετε αυτά τα λάθη, τότε αυτή είναι η διαδικασία για να υπολογίζετε τις πιθανές αβεβαιότητες των παραμέτρων  $\alpha$  και  $\beta$ . Θέτουμε  $\sigma_i \equiv 1$  μέσα σε όλες τις εξισώσεις και πολλαπλασιάζουμε τα  $\sigma_\alpha$  και  $\sigma_\beta$ , όπως προκύπτουν από τις εξισώσεις, με τον επιπλέον παράγοντα  $\sqrt{x^2/(N-2)}$  όπου το  $x^2$  υπολογίζεται από την εξίσωση  $x$  με τους καταλλήλους παράγοντες  $\alpha$  και  $\beta$ . Η παραπάνω διαδικασία είναι ισοδύναμη για να συμπεράνουμε κατά πόσο ταιριάζουν τα δεδομένα στο μοντέλο, ώστε να μην πάρουμε ανεξάρτητα δεδομένα πιθανότητας  $Q$ .

Μια σχέση μεταξύ των γραμμικής συσχέτισης συντελεστών και του υπολογισμού καταλληλότητας  $X^2$ . Για δεδομένα χωρίς βάρη (όλα  $\sigma_i=0$ ), η σχέση είναι:

$$x^2 = (1-r^2)N\text{Var}(y_0 \dots y_{n-1})$$

όπου

$$N\text{Var}(y_0 \dots y_{N-1}) \equiv \sum_{i=0}^{N-1} (y_i - \bar{y})^2$$

Για δεδομένα με διάφορα βάρη  $\sigma_i$ , οι παραπάνω εξισώσεις ισχύουν αν το σύνολο των εξισώσεων βαραίνει κατά  $1/\sigma_i^2$ .

Η παρακάτω συνάρτηση, κατάλληλη, εκτελεί τις διαδικασίες που έχουμε αναφέρει. Όταν τα βάρη είναι γνωστά, οι υπολογισμοί ανταποκρίνονται στους τύπους παραπάνω. Αν όμως τα βάρη δεν δίνονται, τότε η ρουτίνα θεωρεί ίδιες τιμές βαρών για κάθε σημείο και υποθέτει ότι οι τιμές που υπολογίστηκαν είναι κατάλληλες.

Οι τύποι είναι ευαίσθητοι σε σφάλματα στρογγυλοποίησης. Συνεπώς, τις ξαναγράφουμε ως ακολούθως:

$$t_i = \frac{1}{\sigma_i} \left( x^i - \frac{S_x}{S} \right), \quad i = 0, 1, \dots, N-1$$

και

$$S_u = \sum_{i=0}^{N-1} t_i^2$$

Τότε, μπορεί να επιβεβαιωθεί με άμεση αντικατάσταση,

$$b = \frac{1}{S_u} \sum_{i=0}^{N-1} \frac{t_i y_i}{\sigma_i}$$

$$a = \frac{S_y - S_x b}{S}$$

$$\sigma_a^2 = \frac{1}{S} \left( 1 + \frac{S_x^2}{SS_u} \right)$$

$$\sigma_b^2 = \frac{1}{S_u}$$

$$\text{Cov}(a, b) = -\frac{S_x}{SS_u}$$

$$r_{ab} = \frac{\text{Cov}(a, b)}{\sigma_a \sigma_b}$$

# Κεφάλαιο 2ο

**ΛΕΙΤΟΥΡΓΙΑ & ΧΡΗΣΗ  
ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ**

## 2.1 Εργαλεία προγραμματισμού

Το πρόγραμμα δημιουργήθηκε σε περιβάλλον C++ **Visual** με το πρόγραμμα C++ **Builder** έκδοση 6. Το περιβάλλον διεπαφής του χρήστη είναι πλήρως οπτικό και αποτελείται από μία σειρά φορμών που σχεδιάστηκαν με το παραπάνω πρόγραμμα.

Οι εικόνες που χρησιμοποιούνται στις φόρμες σχεδιάστηκαν με το πρόγραμμα επεξεργασίας εικόνας **Adobe Photoshop CS**.

## 2.2 Περιγραφή προγράμματος

Το πρόγραμμα «Αριθμητικές μέθοδοι» έχει ως σκοπό τον υπολογισμό συνθέτων αριθμητικών μεθόδων, με δεδομένα που εισάγει ο χρήστης, σε πραγματικό χρόνο. Επίσης δίνεται η δυνατότητα υπολογισμού γραφικής παράστασης σε ορισμένες μεθόδους.

Οι μέθοδοι χωρίζονται σε σύνολα ασκήσεων, που αποτελούνται από ομάδες φορμών η κάθε άσκηση, που περιέχουν τις μεθόδους και τις επιλογές τους.

Επίσης υπάρχει πλήρης βοήθεια για την χρήση του προγράμματος και των μεθόδων που υλοποιείται επίσης από φόρμες.

### 2.2.1 Φόρμες

Οι φόρμες που χρησιμοποιούνται για την αλληλεπίδραση του χρήστη με το πρόγραμμα είναι οι ακόλουθες:

- Εισαγωγική φόρμα
- Φόρμα ασκήσεων
- Φόρμα επιλογής μεθόδων εύρεσης ριζών
- Φόρμα επιλογής μεθόδων επίλυσης συστημάτων
- Φόρμα επιλογής συναρτήσεων
- Φόρμα υπολογισμού μεθόδου διχοτόμησης
- Φόρμα υπολογισμού μεθόδου τέμνουσας
- Φόρμα υπολογισμού μεθόδου regula falsi
- Φόρμα υπολογισμού μεθόδου του Newton
- Φόρμα υπολογισμού μεθόδου Jacobi

- Φόρμα υπολογισμού μεθόδου Gauss - Seidel
- Φόρμα υπολογισμού μεθόδου ελαχίστων τετραγώνων
- Φόρμα υπολογισμού αποδοτικότητας μεθόδου
- Φόρμα βοήθειας χρήσης του προγράμματος
- Φόρμα βοήθειας μεθόδων
- Φόρμα σχετικά με το πρόγραμμα

Στις παρακάτω σελίδες ακολουθεί περιγραφή και ανάλυση των φορμών που χρησιμοποιούνται:

## ΕΙΣΑΓΩΓΙΚΗ ΦΟΡΜΑ



Εικόνα 2.2.1

Η εισαγωγική φόρμα υλοποιείται από μία εικόνα με το λογότυπο του προγράμματος και από ένα χρονόμετρο (Εικόνα 2.2.1).

Το χρονόμετρο κρατά εμφανή την παρούσα εικόνα για κάποια δευτερόλεπτα και έπειτα εμφανίζει την φόρμα ασκήσεων.

Ο χρήστης δεν μπορεί να εκτελέσει καμία ενέργεια στην παρούσα φόρμα, γι' αυτό και εμφανίζεται ο δείκτης αναμονής στον δείκτη του ποντικιού.

## ΦΟΡΜΑ ΑΣΚΗΣΕΩΝ

Η φόρμα ασκήσεων υλοποιείται από πέντε Bit Buttons με εικονίδια και μια εικόνα - φόντο της φόρμας (Εικόνα 2.2.2).

Ο χρήστης μπορεί να επιλέξει μία από τις διαθέσιμες ασκήσεις που επιθυμεί. Το πλήκτρο «Βοήθεια» εμφανίζει την φόρμα βοήθειας χρήσης και το πλήκτρο «Έξοδος» τερματίζει το πρόγραμμα.

Τα πλήκτρα ασκήσεων είναι 3:

- «Εύρεση ριζών» που οδηγεί στη φόρμα επιλογής μεθόδων εύρεσης ριζών (διχοτόμηση, τέμνουσα, regula falsi, Newton).
- «Επίλυση συστημάτων» που οδηγεί στη φόρμα επιλογής μεθόδων επίλυσης (Jacobi, Gauss – Seidel).
- «Σύστημα ελαχίστων τετραγώνων» που οδηγεί στην φόρμα υπολογισμού της μεθόδου ελαχίστου τετράγωνου.



Εικόνα 2.2.2

## ΦΟΡΜΑ ΕΠΙΛΟΓΗΣ ΜΕΘΟΔΩΝ ΕΥΡΕΣΗΣ ΡΙΖΩΝ

Η φόρμα επιλογής μεθόδων εύρεσης ριζών υλοποιείται από 6 Bit Buttons με εικονίδια και από μία εικόνα - επικεφαλίδα της φόρμας (Εικόνα 2.2.3).

Τέσσερα από τα πλήκτρα οδηγούν στις μεθόδους εύρεσης ριζών:

- «*Διχοτόμησης*», που οδηγεί στην φόρμα υπολογισμού της μεθόδου διχοτόμησης.
- «*Τέμνουσας*», που οδηγεί στην φόρμα υπολογισμού της μεθόδου τέμνουσας.
- «*Regula falsi*», που οδηγεί στην φόρμα υπολογισμού της μεθόδου regula falsi.
- «*Newton*», που οδηγεί στην φόρμα υπολογισμού της μεθόδου του Newton.

Το πλήκτρο «*Ασκήσεις*» οδηγεί στην φόρμα ασκήσεων και το πλήκτρο «*Έξοδος*» τερματίζει το πρόγραμμα.



Εικόνα 2.2.3

## ΦΟΡΜΑ ΕΠΙΛΟΓΗΣ ΜΕΘΟΔΩΝ ΕΠΙΛΥΣΗΣ ΣΥΣΤΗΜΑΤΩΝ

Η φόρμα επιλογής μεθόδων επίλυσης συστημάτων υλοποιείται από 4 Bit Buttons και την εικόνα – επικεφαλίδα της φόρμας (Εικόνα 2.2.4).

Δύο από τα πλήκτρα οδηγούν στις μεθόδους επίλυσης συστήματος:

- «*Jacobi*», που οδηγεί στη φόρμα υπολογισμού της μεθόδου *Jacobi*
- «*Gauss-Seidel*», που οδηγεί στη φόρμα υπολογισμού της μεθόδου *Gauss - Seidel*.

Το πλήκτρο «*Ασκήσεις*» οδηγεί στην φόρμα ασκήσεων και το πλήκτρο «*Έξοδος*» τερματίζει το πρόγραμμα.



Εικόνα 2.2.4

## ΦΟΡΜΑ ΣΥΝΑΡΤΗΣΕΩΝ

Η φόρμα συναρτήσεων εμφανίζεται από οποιαδήποτε μέθοδο εύρεσης ριζών από το μενού **Μέθοδος** → «**Εισαγωγή συνάρτησης**».



Η φόρμα υλοποιείται από 1 List Box, 1 Button, 8 Labels, 3 Edit Boxes, 1 χρονόμετρο και από 2 Bit Buttons (Εικόνα 2.2.5).

Το List Box περιέχει μια ομάδα συναρτήσεων, από την οποία επιλέγει ο χρήστης μια και έπειτα επιλέγει το πλήκτρο «επιλογή τύπου».

Τα 4 από τα 8 Labels εμφανίζουν την μορφή κάθε συνάρτησης.

Το Label «αριθμοί έτοιμοι» δείχνει την κατάσταση επιλογής συνάρτησης όταν έχει επιλεγθεί το πλήκτρο επιβεβαίωσης (OK).

Αν είναι ενεργό τότε σημαίνει ότι η συνάρτηση και οι αριθμοί είναι έτοιμοι για χρήση στη μέθοδο.

Τα υπόλοιπα Labels περιγράφουν τα Edit Boxes.

Το πλήκτρο «επιλογή τύπου» ελέγχει για επιλεγμένη συνάρτηση και ενεργοποιεί το κατάλληλο Label, που εμφανίζει την μορφή της συνάρτησης.

Το πλήκτρο «OK» ελέγχει και κλείνει τη φόρμα, επιβεβαιώνοντας τις επιλογές του χρήστη.

Το πλήκτρο «Cancel» κλείνει τη φόρμα διαγράφοντας τις επιλογές του χρήστη.

Τα Edit Boxes δέχονται τις τιμές της συνάρτησης (a,b,c). Το τρίτο Edit Box εμφανίζεται μόνο στην περίπτωση της επιλογής της συνάρτησης τετραγώνου. Οι αποδεκτές τιμές των Edit είναι πραγματικοί αριθμοί.

Οι επιλογές του χρήστη γίνονται αποδεκτές μόνο αν έχουν οριστεί συνάρτηση και τιμές της συνάρτησης.

Εικόνα 2.2.5

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ ΔΙΧΟΤΟΜΗΣΗΣ

Η φόρμα υπολογισμού μεθόδου διχοτόμησης υλοποιείται από 18 Labels, 11 Edit Box, 6 Buttons, 2 Check Boxes, 2 Radio Buttons, ένα κεντρικό μενού και από ένα Image με Labels που αναπαριστά την γραφική παράσταση (Εικόνα 2.2.6)

Το κεντρικό μενού αποτελείται από 3 κατηγορίες:

- **Μέθοδος** με επιλογές: «εισαγωγή συνάρτησης», «αποδοτικότητα μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Γραφική** με επιλογές: «εμφάνιση γραφικής» και «επιλογές γραφικής»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «**Εισαγωγή συνάρτησης**» οδηγεί στην φόρμα συναρτήσεων.

Το πλήκτρο **Μέθοδος** → «**Αποδοτικότητα μεθόδου**» οδηγεί στην φόρμα αποδοτικότητας μεθόδου.

Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων εύρεσης ριζών.

Το πλήκτρο **Μέθοδος** → «**Έξοδος**» τερματίζει το πρόγραμμα.

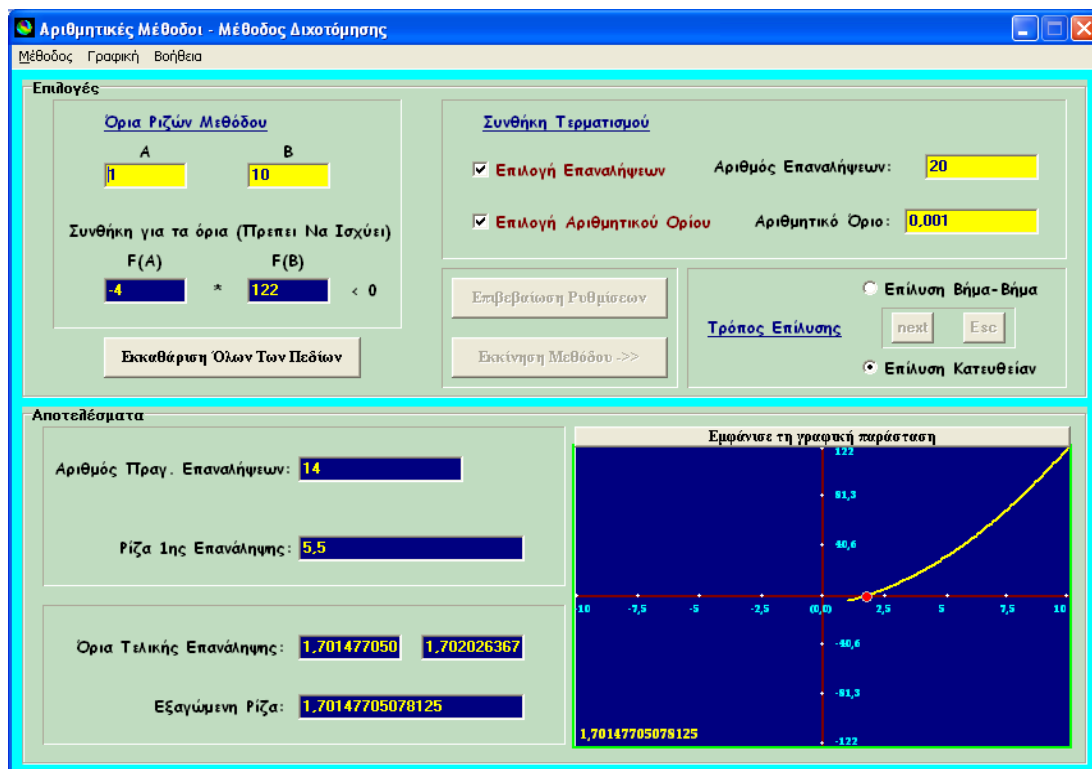
Το πλήκτρο **Γραφική** → «**Εμφάνιση γραφικής**» ορίζει αν η γραφική θα εμφανίζεται ή όχι.

Το πλήκτρο **Γραφική** → «**Επιλογές γραφικής**» υποδιαιρείται σε 5 επιλογές: «λύση», «γραφική παράσταση», «άξονες», «φόντο» και «σημεία» που εμφανίζουν μία παλέτα χρωμάτων για αλλαγή του επιλεγμένου στοιχείου.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.



Εικόνα 2.2.6

Στα 4 από τα 11 Edit Boxes ο χρήστης μπορεί να εισάγει δεδομένα για την εκτέλεση της μεθόδου.

Στα Edit A και B ο χρήστης μπορεί να ορίσει τις τιμές του ορίου, στο οποίο θα υπολογιστεί η ρίζα. Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Στο Edit «αριθμός επαναλήψεων» ο χρήστης εισάγει τις επαναλήψεις της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί.

Στο Edit «αριθμητικό όριο» ο χρήστης εισάγει την ακρίβεια υπολογισμού της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί πραγματικοί αριθμοί κάτω από 1.

Στα Edit F(A) και F(B) υπολογίζονται οι συναρτήσεις των ορίων και ελέγχεται αν ισχύει η ανίσωση  $F(A) \cdot F(B) < 0$  από τον χρήστη. Η ανίσωση πρέπει να ισχύει για να εκτελεστεί η μέθοδος.

Στα υπόλοιπα Edit εμφανίζονται τα αποτελέσματα και οι επαναλήψεις της μεθόδου.

Τα Check Box χρησιμοποιούνται για την ενεργοποίηση των επιλογών «αριθμός επαναλήψεων» και «αριθμητικό όριο».

Τα Radio Button χρησιμοποιούνται για την επιλογή του τρόπου εκτέλεσης της μεθόδου. Οι δυνατές επιλογές είναι η απευθείας εκτέλεση και η εκτέλεση βήμα-βήμα. Στην εκτέλεση βήμα - βήμα η μέθοδος δίνει αποτελέσματα για κάθε επανάληψη.

Το πλήκτρο «εκκαθάριση όλων των πεδίων» καθαρίζει όλα τα edit box που ο χρήστης δίνει δεδομένα.

Το πλήκτρο «επιβεβαίωση ρυθμίσεων» ελέγχει αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά και υπολογίζει τις συναρτήσεις των ορίων.

Η επιλογή του παραπάνω πλήκτρου είναι απαραίτητη για την επιλογή «εκτέλεση μεθόδου».

Το πλήκτρο «εκτέλεση μεθόδου» μετατρέπει τα δεδομένα σε αριθμούς και υπολογίζει τη ρίζα με τον τύπο της διχοτόμησης (Κεφ. 1.2.1) για τις επαναλήψεις που ορίστηκαν από το χρήστη. Αν η ρίζα υπολογιστεί με τη διδόμενη ακρίβεια για τις επαναλήψεις που επιλέχθηκαν τότε εμφανίζεται ένας διάλογος με την παρατήρηση «Η μέθοδος συγκλίνει» και εμφανίζονται τα αποτελέσματα και η γραφική παράσταση. Σε αντίθετη περίπτωση τα αποτελέσματα και η γραφική δεν εμφανίζονται και εμφανίζεται ένας διάλογος με το κείμενο «Η μέθοδος δε συγκλίνει». Αυτό ισχύει για την απευθείας εκτέλεση της μεθόδου, στην εκτέλεση Βήμα - Βήμα όμως η μέθοδος εκτελείται μόνο για την πρώτη επανάληψη και εμφανίζονται αποτελέσματα. Για την επόμενη επανάληψη ο χρήστης επιλέγει το πλήκτρο «next» που τώρα είναι ενεργό. Για πρόωρο τερματισμό της εκτέλεσης Βήμα - Βήμα της μεθόδου, ο χρήστης πρέπει να επιλέξει το πλήκτρο «Esc».

Κατά την εκτέλεση της μεθόδου, σε κάθε επανάληψη υπολογίζεται το μέσο του διαστήματος ( $x_i$ ) του ορίου που δόθηκε. Υπολογίζονται επίσης και οι συναρτήσεις του διαστήματος  $\alpha$  και  $x_i$ . Έπειτα, αν ισχύει η ανίσωση  $F(\alpha) \cdot F(x_i) < 0$  τότε τα νέα όρια είναι  $(\alpha, x_i)$ . Σε αντίθετη περίπτωση είναι  $(x_i, \beta)$ .

Η γραφική παράσταση της εξίσωσης υλοποιείται από μία εικόνα με επιμέρους Labels και την εκτέλεση συναρτήσεων **Move To** και **Line To**, βασισμένα στα αποτελέσματα της μεθόδου.

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ ΤΕΜΝΟΥΣΑΣ

Η φόρμα υπολογισμού μεθόδου τέμνουσας υλοποιείται από 18 Labels, 10 Edit Boxes, 6 Buttons, 2 Check Boxes, 2 Radio Buttons, ένα κεντρικό μενού και από ένα Image με labels που αναπαριστά την γραφική παράσταση (Εικόνα 2.2.7)

Το κεντρικό μενού αποτελείται από 3 κατηγορίες:

- **Μέθοδος** με επιλογές: «εισαγωγή συνάρτησης», «αποδοτικότητα μεθόδου», «Επιστροφή στο μενού επιλογών» και «έξοδος»
- **Γραφική** με επιλογές: «εμφάνιση γραφικής» και «επιλογές γραφικής»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «Εισαγωγή συνάρτησης» οδηγεί στην φόρμα συναρτήσεων.

Το πλήκτρο **Μέθοδος** → «Αποδοτικότητα μεθόδου» οδηγεί στην φόρμα αποδοτικότητας μεθόδου.

Το πλήκτρο **Μέθοδος** → «Επιστροφή στο μενού επιλογών» οδηγεί στην φόρμα επιλογής μεθόδων εύρεσης ριζών.

Το πλήκτρο **Μέθοδος** → «Έξοδος» τερματίζει το πρόγραμμα.

Το πλήκτρο **Γραφική** → «Εμφάνιση γραφικής» ορίζει αν η γραφική θα εμφανίζεται ή όχι.

Το πλήκτρο **Γραφική** → «Επιλογές γραφικής» υποδιαιρείται σε 5 επιλογές: «λύση», «γραφική παράσταση», «άξονες», «φόντο» και «σημεία» που εμφανίζουν μία παλέτα χρωμάτων για αλλαγή του επιλεγμένου στοιχείου.

Το πλήκτρο **Βοήθεια** → «Βοήθεια χρήσης» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «Βοήθεια μεθόδου» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «Σχετικά με» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.



Εικόνα 2.2.7

Στα 4 από τα 10 Edit Box ο χρήστης μπορεί να εισάγει δεδομένα για την εκτέλεση της μεθόδου.

Στα Edit A και B ο χρήστης μπορεί να ορίσει τις τιμές του ορίου, στο οποίο θα υπολογιστεί η ρίζα. Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Στο Edit «αριθμός επαναλήψεων» ο χρήστης εισάγει τις επαναλήψεις της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί.

Στο Edit «αριθμητικό όριο» ο χρήστης εισάγει την ακρίβεια υπολογισμού της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί πραγματικοί αριθμοί κάτω από 1.

Στα Edit F(A) και F(B) υπολογίζονται οι συναρτήσεις των ορίων και ελέγχεται αν ισχύει η ανίσωση  $F(A) \cdot F(B) < 0$  από τον χρήστη. Η ανίσωση πρέπει να ισχύει για να εκτελεστεί η μέθοδος.

Στα υπόλοιπα Edit εμφανίζονται τα αποτελέσματα και οι επαναλήψεις της μεθόδου.

Τα Check Box χρησιμοποιούνται για την ενεργοποίηση των επιλογών «αριθμός επαναλήψεων» και «αριθμητικό όριο».

Τα Radio Buttons χρησιμοποιούνται για την επιλογή του τρόπου εκτέλεσης της μεθόδου. Οι δυνατές επιλογές είναι η απευθείας εκτέλεση και η εκτέλεση βήμα - βήμα. Στην εκτέλεση βήμα - βήμα η μέθοδος δίνει αποτελέσματα για κάθε επανάληψη.

Το πλήκτρο «εκκαθάριση όλων των πεδίων» καθαρίζει όλα τα Edit Boxes που ο χρήστης δίνει δεδομένα.

Το πλήκτρο «επιβεβαίωση ρυθμίσεων» ελέγχει αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά και υπολογίζει τις συναρτήσεις των ορίων.

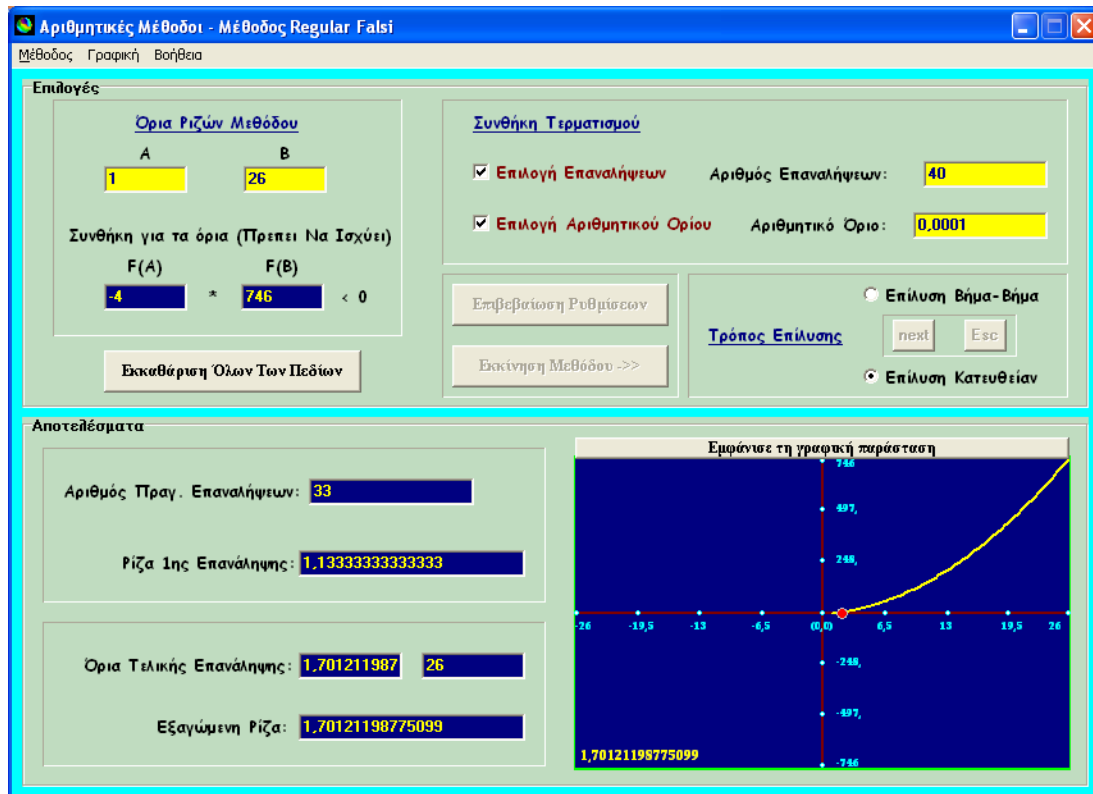
Η επιλογή του παραπάνω πλήκτρου είναι απαραίτητη για την επιλογή «εκτέλεση μεθόδου».

Το πλήκτρο «εκτέλεση μεθόδου» μετατρέπει τα δεδομένα σε αριθμούς και υπολογίζει τη ρίζα με τον τύπο της τέμνουσας (Κεφ. 1.2.2) για τις επαναλήψεις που ορίστηκαν από το χρήστη. Αν η ρίζα υπολογιστεί με τη διδόμενη ακρίβεια για τις επαναλήψεις που επιλέχθηκαν τότε εμφανίζεται ένας διάλογος με την παρατήρηση «Η μέθοδος συγκλίνει» και εμφανίζονται τα αποτελέσματα και η γραφική παράσταση. Σε αντίθετη περίπτωση τα αποτελέσματα και η γραφική δεν εμφανίζονται και εμφανίζεται ένας διάλογος με το κείμενο «Η μέθοδος δε συγκλίνει». Αυτό ισχύει για την απευθείας εκτέλεση της μεθόδου, στην εκτέλεση Βήμα - Βήμα όμως η μέθοδος εκτελείται μόνο για την πρώτη επανάληψη και εμφανίζονται αποτελέσματα. Για την επόμενη επανάληψη ο χρήστης επιλέγει το πλήκτρο «next» που τώρα είναι ενεργό. Για πρόωρο τερματισμό της εκτέλεσης Βήμα - Βήμα της μεθόδου, ο χρήστης πρέπει να επιλέξει το πλήκτρο «Esc».

Η γραφική της εξίσωσης υλοποιείται από μία εικόνα με επιμέρους Labels και την εκτέλεση συναρτήσεων **Move To** και **Line To**, βασισμένα στα αποτελέσματα της μεθόδου.

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ REGULA FALSI

Η φόρμα υπολογισμού μεθόδου regula falsi υλοποιείται από 18 Labels, 11 Edit Boxes, 6 Buttons, 2 Check Boxes, 2 Radio Buttons, ένα κεντρικό μενού και από ένα Image με Labels που αναπαριστά την γραφική παράσταση (Εικόνα 2.2.8)



Εικόνα 2.2.8

Το κεντρικό μενού αποτελείται από 3 κατηγορίες:

- **Μέθοδος** με επιλογές: «εισαγωγή συνάρτησης», «αποδοτικότητα μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Γραφική** με επιλογές: «εμφάνιση γραφικής» και «επιλογές γραφικής»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «Εισαγωγή συνάρτησης» οδηγεί στην φόρμα συναρτήσεων.

Το πλήκτρο **Μέθοδος** → «Αποδοτικότητα μεθόδου» οδηγεί στην φόρμα αποδοτικότητας μεθόδου.



Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων εύρεσης ριζών.

Το πλήκτρο **Μέθοδος** → «**Εξόδος**» τερματίζει το πρόγραμμα.

Το πλήκτρο **Γραφική** → «**Εμφάνιση γραφικής**» ορίζει αν η γραφική θα εμφανίζεται ή όχι.

Το πλήκτρο **Γραφική** → «**Επιλογές γραφικής**» υποδιαιρείται σε 5 επιλογές: «**λύση**», «**γραφική παράσταση**», «**άξονες**», «**φόντο**» και «**σημεία**» που εμφανίζουν μία παλέτα χρωμάτων για αλλαγή του επιλεγμένου στοιχείου.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.

Στα 4 από τα 11 Edit Boxes ο χρήστης μπορεί να εισάγει δεδομένα για την εκτέλεση της μεθόδου.

Στα Edit A και B ο χρήστης μπορεί να ορίσει τις τιμές του ορίου, στο οποίο θα υπολογιστεί η ρίζα. Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Στο Edit «**αριθμός επαναλήψεων**» ο χρήστης εισάγει τις επαναλήψεις της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί.

Στο Edit «**αριθμητικό όριο**» ο χρήστης εισάγει την ακρίβεια υπολογισμού της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί πραγματικοί αριθμοί κάτω από 1.

Στα Edit F(A) και F(B) υπολογίζονται οι συναρτήσεις των ορίων και ελέγχεται αν ισχύει η ανίσωση  $F(A) \cdot F(B) < 0$  από τον χρήστη. Η ανίσωση πρέπει να ισχύει για να εκτελεστεί η μέθοδος.

Στα υπόλοιπα Edit εμφανίζονται τα αποτελέσματα και οι επαναλήψεις της μεθόδου.

Τα Check Boxes χρησιμοποιούνται για την ενεργοποίηση των επιλογών «**αριθμός επαναλήψεων**» και «**αριθμητικό όριο**».

Τα Radio Buttons χρησιμοποιούνται για την επιλογή του τρόπου εκτέλεσης της μεθόδου. Οι δυνατές επιλογές είναι η απευθείας εκτέλεση και η εκτέλεση βήμα - βήμα. Στην εκτέλεση βήμα - βήμα, η μέθοδος δίνει αποτελέσματα για κάθε επανάληψη.

Το πλήκτρο «εκκαθάριση όλων των πεδίων» καθαρίζει όλα τα Edit boxes που ο χρήστης δίνει δεδομένα.

Το πλήκτρο «επιβεβαίωση ρυθμίσεων» ελέγχει αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά και υπολογίζει τις συναρτήσεις των ορίων.

Η επιλογή του παραπάνω πλήκτρου είναι απαραίτητη για την επιλογή «εκτέλεση μεθόδου».

Το πλήκτρο «εκτέλεση μεθόδου» μετατρέπει τα δεδομένα σε αριθμούς και υπολογίζει τη ρίζα με τον τύπο της regula falsi (Κεφ 1.2.2) για τις επαναλήψεις που ορίστηκαν από το χρήστη. Αν η ρίζα υπολογιστεί με τη διδόμενη ακρίβεια για τις επαναλήψεις που επιλέχθηκαν τότε εμφανίζεται ένας διάλογος με την παρατήρηση «Η μέθοδος συγκλίνει» και εμφανίζονται τα αποτελέσματα και η γραφική παράσταση. Σε αντίθετη περίπτωση τα αποτελέσματα και η γραφική δεν εμφανίζονται και εμφανίζεται ένας διάλογος με το κείμενο «Η μέθοδος δε συγκλίνει». Αυτό ισχύει για την απευθείας εκτέλεση της μεθόδου, στην εκτέλεση Βήμα - Βήμα όμως η μέθοδος εκτελείται μόνο για την πρώτη επανάληψη και εμφανίζονται αποτελέσματα. Για την επόμενη επανάληψη ο χρήστης επιλέγει το πλήκτρο «next» που τώρα είναι ενεργό. Για πρόωρο τερματισμό της εκτέλεσης Βήμα - Βήμα της μεθόδου ο χρήστης πρέπει να επιλέξει το πλήκτρο «Esc».

Κατά την εκτέλεση της μεθόδου, σε κάθε επανάληψη υπολογίζονται οι συναρτήσεις του διαστήματος  $\alpha$  και  $\chi_i$ . Έπειτα, αν ισχύει η ανίσωση  $F(\alpha) \cdot F(\chi_i) < 0$  τότε τα νέα όρια είναι  $(\alpha, \chi_i)$ . Σε αντίθετη περίπτωση είναι  $(\chi_i, \beta)$ .

Η γραφική της εξίσωσης υλοποιείται από μία εικόνα με επιμέρους Labels και την εκτέλεση συναρτήσεων **Move To** και **Line To**, βασισμένα στα αποτελέσματα της μεθόδου.

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ ΤΟΥ NEWTON

Η φόρμα υπολογισμού μεθόδου του Newton υλοποιείται από 18 Labels, 10 Edit Boxes, 6 Buttons, 2 Check Boxes, 2 Radio Buttons, ένα κεντρικό μενού και από ένα Image με Labels που αναπαριστά τη γραφική παράσταση (Εικόνα 2.2.9)



Εικόνα 2.2.9

Το κεντρικό μενού αποτελείται από 3 κατηγορίες:

- **Μέθοδος** με επιλογές: «εισαγωγή συνάρτησης», «αποδοτικότητα μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Γραφική** με επιλογές: «εμφάνιση γραφικής» και «επιλογές γραφικής»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «Εισαγωγή συνάρτησης» οδηγεί στην φόρμα συναρτήσεων.

Το πλήκτρο **Μέθοδος** → «Αποδοτικότητα μεθόδου» οδηγεί στην φόρμα αποδοτικότητας μεθόδου.

Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων εύρεσης ριζών.

Το πλήκτρο **Μέθοδος** → «**Εξοδος**» τερματίζει το πρόγραμμα.

Το πλήκτρο **Γραφική** → «**Εμφάνιση γραφικής**» ορίζει αν η γραφική θα εμφανίζεται ή όχι.

Το πλήκτρο **Γραφική** → «**Επιλογές γραφικής**» υποδιαιρείται σε 5 επιλογές: «**λύση**», «**γραφική παράσταση**», «**άξονες**», «**φόντο**» και «**σημεία**» που εμφανίζουν μία παλέτα χρωμάτων για αλλαγή του επιλεγμένου στοιχείου.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.

Στα 3 από τα 10 Edit Boxes ο χρήστης μπορεί να εισάγει δεδομένα για την εκτέλεση της μεθόδου.

Στο Edit  $X_0$  ο χρήστης μπορεί να ορίσει μια αρχική τιμή, βάσει της οποίας θα υπολογιστεί η ρίζα. Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Στο Edit «**αριθμός επαναλήψεων**» ο χρήστης εισάγει τις επαναλήψεις της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί.

Στο Edit «**αριθμητικό όριο**» ο χρήστης εισάγει την ακρίβεια υπολογισμού της μεθόδου. Οι αποδεκτές τιμές είναι μη αρνητικοί πραγματικοί αριθμοί κάτω από 1.

Στα υπόλοιπα Edits εμφανίζονται τα αποτελέσματα και οι επαναλήψεις της μεθόδου.

Τα Check Boxes χρησιμοποιούνται για την ενεργοποίηση των επιλογών «**αριθμός επαναλήψεων**» και «**αριθμητικό όριο**».

Τα Radio Buttons χρησιμοποιούνται για την επιλογή του τρόπου εκτέλεσης της μεθόδου. Οι δυνατές επιλογές είναι η απευθείας εκτέλεση και η εκτέλεση βήμα - βήμα. Στην εκτέλεση βήμα - βήμα η μέθοδος δίνει αποτελέσματα για κάθε επανάληψη.

Το πλήκτρο «**τυχαία**» δίνει μια τυχαία τιμή από το 1 μέχρι το 10 σαν αρχική τιμή της ρίζας.

Το πλήκτρο «**εκκαθάριση όλων των πεδίων**» καθαρίζει όλα τα Edit Boxes που ο χρήστης δίνει δεδομένα.

Το πλήκτρο «επιβεβαίωση ρυθμίσεων» ελέγχει αν τα στοιχεία που έδωσε ο χρήστης είναι σωστά και υπολογίζει τις συναρτήσεις των ορίων.

Η επιλογή του παραπάνω πλήκτρου είναι απαραίτητη για την επιλογή «εκτέλεση μεθόδου».

Το πλήκτρο «εκτέλεση μεθόδου» μετατρέπει τα δεδομένα σε αριθμούς και υπολογίζει τη ρίζα με τον τύπο της Newton (Κεφ 1.2.3) για τις επαναλήψεις που ορίστηκαν από το χρήστη. Αν η ρίζα υπολογιστεί με τη διδόμενη ακρίβεια για τις επαναλήψεις που επιλέχθηκαν τότε εμφανίζεται ένας διάλογος με την παρατήρηση «*Η μέθοδος συγκλίνει*» και εμφανίζονται τα αποτελέσματα και η γραφική παράσταση. Σε αντίθετη περίπτωση τα αποτελέσματα και η γραφική δεν εμφανίζονται και εμφανίζεται ένας διάλογος με το κείμενο «*Η μέθοδος δε συγκλίνει*». Αυτό ισχύει για την απευθείας εκτέλεση της μεθόδου, στην εκτέλεση Βήμα - Βήμα όμως η μέθοδος εκτελείται μόνο για την πρώτη επανάληψη και εμφανίζονται αποτελέσματα. Για την επόμενη επανάληψη ο χρήστης επιλέγει το πλήκτρο «next» που τώρα είναι ενεργό. Για πρόωρο τερματισμό της εκτέλεσης Βήμα - Βήμα της μεθόδου ο χρήστης πρέπει να επιλέξει το πλήκτρο «Esc».

Κατά την εκτέλεση της μεθόδου, σε κάθε επανάληψη υπολογίζονται η συνάρτηση της νέας ρίζας και η παράγωγος της.

Η γραφική της εξίσωσης υλοποιείται από μία εικόνα με επιμέρους Labels και την εκτέλεση συναρτήσεων **Move To** και **Line To**, βασισμένα στα αποτελέσματα της μεθόδου.

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ JACOBI

Η φόρμα υπολογισμού μεθόδου Jacobi υλοποιείται από ένα κεντρικό μενού, 74 Edit Boxes, 9 Labels και 1 Bit Button (Εικόνα 2.2.10).

Το κεντρικό μενού αποτελείται από 2 κατηγορίες:

- **Μέθοδος** με επιλογές: «εκκίνηση μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «**Εκκίνηση μεθόδου**» εκτελείται η μέθοδος .

Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων επίλυσης συστημάτων.

Το πλήκτρο **Μέθοδος** → «**Έξοδος**» τερματίζει το πρόγραμμα.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.

Τα 8 από τα 9 Labels χρησιμοποιούνται για περιγραφή των Edit Boxes, ενώ το ένα Label χρησιμοποιείται ως διακόπτης για την λειτουργία εισαγωγής μεγέθους πίνακα.

Στα 65 από τα 74 Edits ο χρήστης μπορεί να εισάγει δεδομένα.

Τα 63 από τα 65 Edits αναπαριστούν τις τιμές των πινάκων A (δεδομένα), B, X (αρχικών τιμών) και είναι αρχικά απενεργοποιημένα. Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Για να ενεργοποιούν ο χρήστης πρέπει να εισάγει το μέγεθος του πίνακα που επιθυμεί στο Edit «*μέγεθος συστήματος*». Αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί στο όριο (2 με 7) για τα ανάλογα μεγέθη (2X2, 3X3, 4X4, 5X5, 6X6, 7X7).

Στο Edit «*όριο ευκλείδειας νόρμας*» ο χρήστης ορίζει το όριο της ευκλείδειας νόρμας που επιθυμεί.

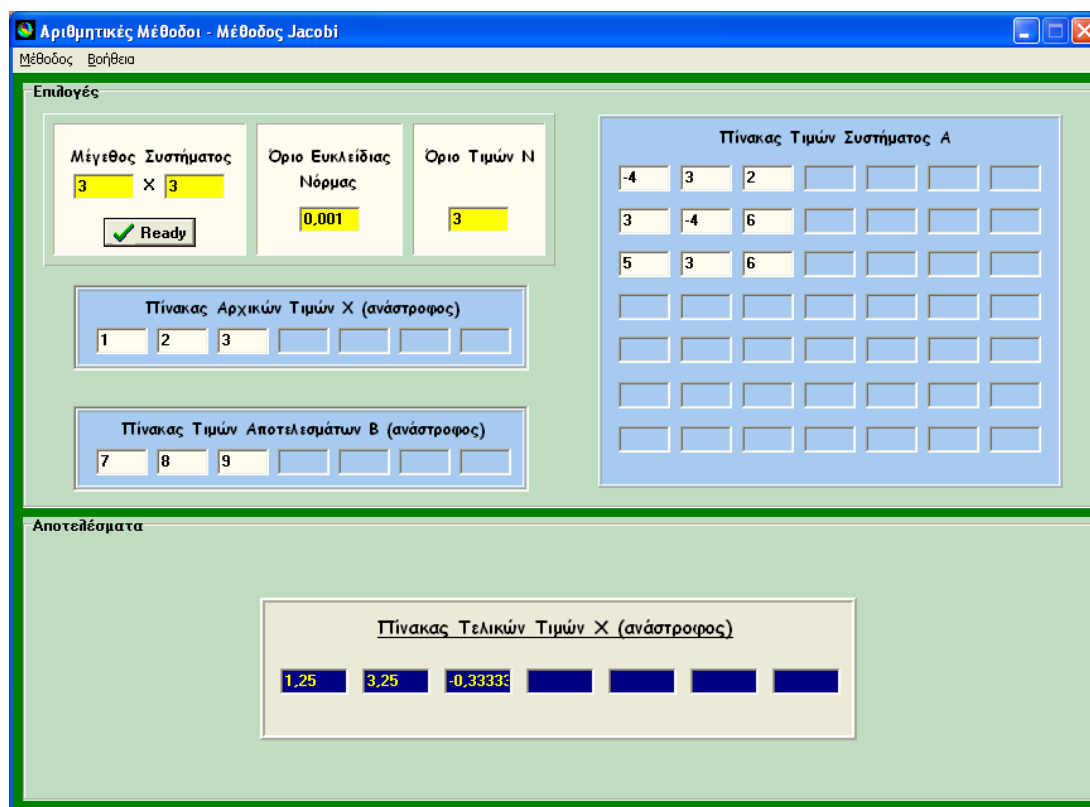
Στο Edit «*όριο τιμών N*» ο χρήστης δεν εισάγει δεδομένα.

Τα υπόλοιπα 7 από τα 74 Edits εμφανίζουν τα αποτελέσματα της μεθόδου.

Το πλήκτρο «OK» χρησιμοποιείται για να επιβεβαιώσει την επιλογή ομάδας του χρήστη και να ενεργοποιήσει τα κατάλληλα πεδία.

Όταν επιβεβαιώνεται με επιτυχία το μέγεθος του συστήματος, το πλήκτρο «OK» αλλάζει σε «Ready».

Κατά την εκτέλεση της μεθόδου, αφού ο χρήστης έχει ορίσει τιμές στους πίνακες και στο όριο της ευκλείδειας νόρμας, το πρόγραμμα μετατρέπει τα στοιχεία σε αριθμούς και υπολογίζει τις ρίζες του συστήματος βάσει του τύπου της Jacobi (Κεφ. 1.2.4).



Εικόνα 2.2.10

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ GAUSS-SEIDEL

Η φόρμα υπολογισμού μεθόδου Gauss - Seidel υλοποιείται από ένα κεντρικό μενού, 74 Edit Boxes, 9 Labels και 1 Bit Button (Εικόνα 2.2.11).

Το κεντρικό μενού αποτελείται από 2 κατηγορίες:

- **Μέθοδος** με επιλογές: «εκκίνηση μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «**Εκκίνηση μεθόδου**» εκτελείται η μέθοδος .

Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων επίλυσης συστημάτων.

Το πλήκτρο **Μέθοδος** → «**Εξοδος**» τερματίζει το πρόγραμμα.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.

Αριθμητικές Μέθοδοι - Μέθοδος Gauss-Seidel

Μέθοδος Βοήθεια

Επιλογές

Μέγεθος Συστήματος: 4 × 4

✓ Ready

Όριο Ευκλείδειας Νόρμας: 0,001

Όριο Τιμών N: 4

Πίνακας Αρχικών Τιμών X (ανάστροφος)

1	2	3	4			
---	---	---	---	--	--	--

Πίνακας Τιμών Αποτελεσμάτων B (ανάστροφος)

7	10	9	10			
---	----	---	----	--	--	--

Πίνακας Τιμών Συστήματος A

3	2	1	4			
-9	2	3	5			
-10	3	4	5			
3	4	4	7			

Αποτελέσματα

Πίνακας Τελικών Τιμών X (ανάστροφος)

5,3333	-33,5	9,04166	17,6904			
--------	-------	---------	---------	--	--	--

Εικόνα 2.2.11



Τα 8 από τα 9 Labels χρησιμοποιούνται για περιγραφή των Edit Boxes, ενώ το ένα Label χρησιμοποιείται ως διακόπτης για την λειτουργία εισαγωγής μεγέθους πίνακα.

Στα 65 από τα 74 Edits ο χρήστης μπορεί να εισάγει δεδομένα.

Τα 63 από τα 65 Edits αναπαριστούν τις τιμές των πινάκων A (δεδομένα), B, X (αρχικών τιμών) και είναι αρχικά απενεργοποιημένα.

Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Για να ενεργοποιούν ο χρήστης πρέπει να εισάγει το μέγεθος του πίνακα που επιθυμεί στο Edit «μέγεθος συστήματος». Αποδεκτές τιμές είναι μη αρνητικοί ακέραιοι αριθμοί στο όριο (2 με 7) για τα ανάλογα μεγέθη (2X2, 3X3, 4X4, 5X5, 6X6, 7X7).

Στο Edit «όριο ευκλείδειας νόρμας» ο χρήστης ορίζει το όριο της ευκλείδειας νόρμας που επιθυμεί.

Στο Edit «όριο τιμών N» ο χρήστης δεν εισάγει δεδομένα.

Τα υπόλοιπα 7 από τα 74 Edits εμφανίζουν τα αποτελέσματα της μεθόδου.

Το πλήκτρο «OK» χρησιμοποιείται για να επιβεβαιώσει την επιλογή ομάδας του χρήστη και να ενεργοποιήσει τα κατάλληλα πεδία.

Όταν επιβεβαιώνεται με επιτυχία το μέγεθος του συστήματος, το πλήκτρο «OK» αλλάζει σε «Ready».

Κατά την εκτέλεση της μεθόδου, αφού ο χρήστης έχει ορίσει τιμές στους πίνακες και στο όριο της ευκλείδειας νόρμας, το πρόγραμμα μετατρέπει τα στοιχεία σε αριθμούς και υπολογίζει τις ρίζες του συστήματος βάσει του τύπου της Gauss - Seidel (Κεφ. 1.2.5).

## ΦΟΡΜΑ ΥΠΟΛΟΓΙΣΜΟΥ ΜΕΘΟΔΟΥ ΕΛΑΧΙΣΤΩΝ ΤΕΤΡΑΓΩΝΩΝ

Η φόρμα υπολογισμού μεθόδου ελαχίστων τετραγώνων υλοποιείται από ένα κεντρικό μενού, 35 Edit Boxes, 1 Page Control με 2 TabSheets, 3 Bit Buttons, 8 Labels και 1 Image με Labels (Εικ. 2.2.12).

Το κεντρικό μενού αποτελείται από 3 κατηγορίες:

- **Μέθοδος** με επιλογές: «εκκίνηση μεθόδου», «επιστροφή στο μενού επιλογών» και «έξοδος»
- **Γραφική** με επιλογές: «εμφάνιση γραφικής» και «επιλογές γραφικής»
- **Βοήθεια** με επιλογές: «βοήθεια χρήσης», «βοήθεια μεθόδου» και «σχετικά με».

Το πλήκτρο **Μέθοδος** → «**Εκκίνηση μεθόδου**» εκτελεί την μέθοδο.

Το πλήκτρο **Μέθοδος** → «**Επιστροφή στο μενού επιλογών**» οδηγεί στην φόρμα επιλογής μεθόδων εύρεσης ριζών.

Το πλήκτρο **Μέθοδος** → «**Εξοδος**» τερματίζει το πρόγραμμα.

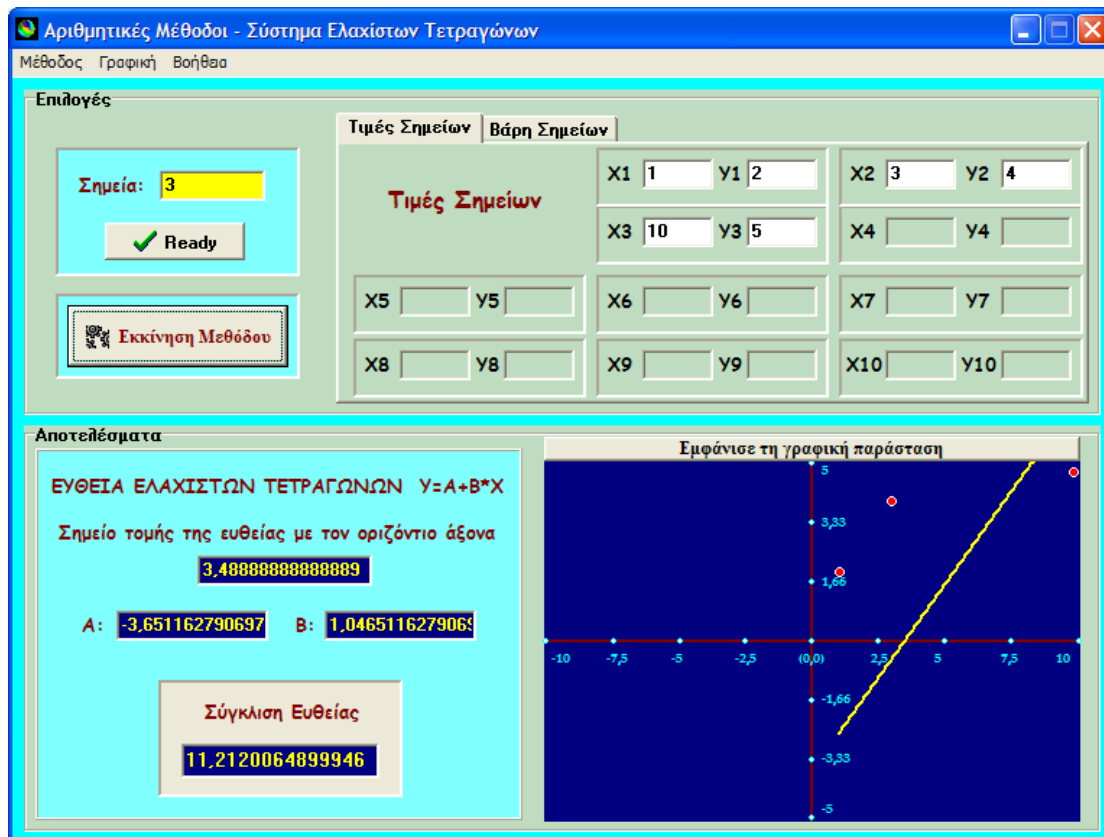
Το πλήκτρο **Γραφική** → «**Εμφάνιση γραφικής**» ορίζει αν η γραφική θα εμφανίζεται ή όχι.

Το πλήκτρο **Γραφική** → «**Επιλογές γραφικής**» υποδιαιρείται σε 5 επιλογές: «λύση», «γραφική παράσταση», «άξονες», «φόντο» και «σημεία» που εμφανίζουν μία παλέτα χρωμάτων για αλλαγή του επιλεγμένου στοιχείου.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια χρήσης**» εμφανίζει τη φόρμα βοήθειας χρήσης.

Το πλήκτρο **Βοήθεια** → «**Βοήθεια μεθόδου**» εμφανίζει τη φόρμα βοήθειας μεθόδων.

Το πλήκτρο **Βοήθεια** → «**Σχετικά με**» εμφανίζει τη φόρμα σχετικά με το πρόγραμμα.



Εικόνα 2.2.12

Το Page Control αποτελείται από δύο TabSheets:

- **Τιμές σημείων** που περιέχει τα Edit των τιμών των σημείων (Εικ. 2.2.13)

Εικόνα 2.2.13

- **Βάρη σημείων** που περιέχει τα Edit των βαρών των σημείων (Εικ. 2.2.14)

Τιμές Σημείων	Βάρη Σημείων	
<b>Βάρη Σημείων</b>	Σημείο 1ο <input type="text" value="2"/>	Σημείο 2ο <input type="text" value="2"/>
	Σημείο 3ο <input type="text" value="2"/>	Σημείο 4ο <input type="text"/>
Σημείο 5ο <input type="text"/>	Σημείο 6ο <input type="text"/>	Σημείο 7ο <input type="text"/>
Σημείο 8ο <input type="text"/>	Σημείο 9ο <input type="text"/>	Σημείο 10ο <input type="text"/>

Εικόνα 2.2.14

Οι αποδεκτές τιμές είναι πραγματικοί αριθμοί.

Στα 21 από τα 35 Edits ο χρήστης μπορεί να εισάγει δεδομένα.

Στο Edit «σημεία» ο χρήστης εισάγει τον αριθμό των σημείων που επιθυμεί. Αποδεκτές τιμές είναι μη-αρνητικοί ακέραιοι αριθμοί .

Έπειτα ο χρήστης επιλέγει το πλήκτρο «OK» για να επιβεβαιώσει τον αριθμό των σημείων. Όταν επιβεβαιώνεται με επιτυχία το μέγεθος του συστήματος, το πλήκτρο «OK» αλλάζει σε «Ready» και ενεργοποιούνται τα ανάλογα Edits στις τιμές σημείων και στα βάρη σημείων.

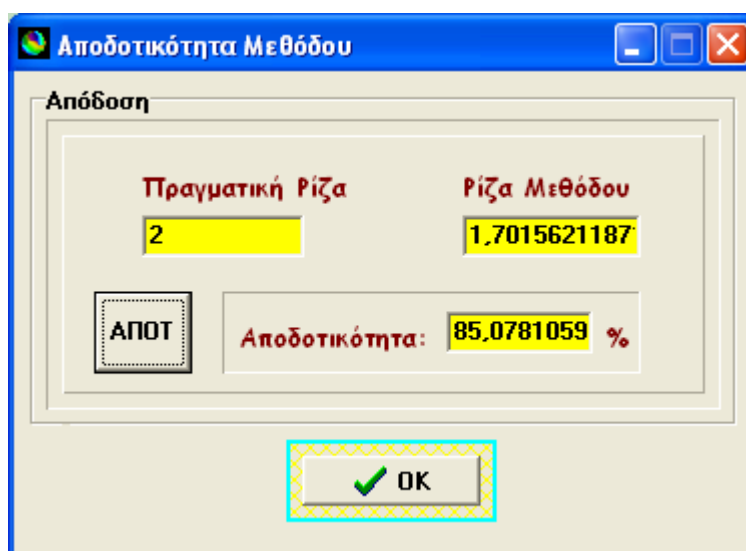
Τα υπόλοιπα Edits χρησιμοποιούνται για την εμφάνιση αποτελεσμάτων.

Το πλήκτρο «εκκίνηση μεθόδου» ελέγχει αν έχουν εισαχθεί σωστά στοιχεία και εκτελεί την μέθοδο. Κατά την εκτέλεση της μεθόδου, το πρόγραμμα μετατρέπει τα στοιχεία που εισήγαγε ο χρήστης σε αριθμούς και υπολογίζει τα  $\alpha$  και  $\beta$  της ευθείας ελαχίστων τετράγωνων, τη σύγκλιση ευθείας και τη γραφική παράσταση (Κεφ. 1.2.6).

Η γραφική της ευθείας υλοποιείται από μία εικόνα με επιμέρους Labels και την εκτέλεση συναρτήσεων **Move To** και **Line To**, βασισμένα στα αποτελέσματα της μεθόδου.

## ΦΟΡΜΑ ΑΠΟΔΟΤΙΚΟΤΗΤΑΣ ΜΕΘΟΔΟΥ

Η φόρμα αποδοτικότητας μεθόδου εμφανίζεται επιλέγοντας το πλήκτρο «αποδοτικότητα μεθόδου» από το μενού **Μέθοδος οποιασδήποτε μεθόδου εύρεσης ρίζας**. Υλοποιείται από 3 Edits, 2 Bit Buttons και 4 Labels (Εικ. 2.2.15)



Εικόνα 2.2.15

Ο χρήστης εισάγει την πραγματική ρίζα στο Edit «πραγματική ρίζα».

Το πλήκτρο «ΑΠΟΤ» υπολογίζει την αποδοτικότητα της μεθόδου από τον τυπο:

$$A = \frac{|X_{\text{πραγ}} - X|}{|X_{\text{πραγ}}|} * 100 \quad (\%)$$

βάσει της ρίζας που έχει υπολογιστεί από την μέθοδο.

Το πλήκτρο «OK» κλείνει τη φόρμα.

## ΦΟΡΜΑ ΒΟΗΘΕΙΑΣ ΧΡΗΣΗΣ ΤΟΥ ΠΡΟΓΡΑΜΜΑΤΟΣ

Η φόρμα βοήθειας χρήσης του προγράμματος εμφανίζεται επιλέγοντας το πλήκτρο «βοήθεια χρήσης» από το μενού **Βοήθεια**.

Η φόρμα υλοποιείται από 6 Bit Buttons, 3 Buttons, 1 memo και μία εικόνα - επικεφαλίδα της φόρμας (Εικ. 2.2.16).

Τα 8 πλήκτρα από τα 9 εμφανίζουν τη βοήθεια για κάθε μέθοδο ή λειτουργία του προγράμματος. Αναλυτικότερα, τα πλήκτρα φορτώνουν αρχεία .txt (text) που περιέχουν την βοήθεια στο memo.

Το πλήκτρο «εισαγωγή» παρουσιάζει την γενική λειτουργία του προγράμματος.

Το πλήκτρο «ασκήσεις» εμφανίζει την βοήθεια για την φόρμα ασκήσεων.

Το πλήκτρο «μέθοδοι» εμφανίζει γενικές πληροφορίες για τις μεθόδους και τρία ακόμη πλήκτρα: «εύρεσης ριζών», «επίλυσης εξισώσεων», «ελαχίστου τετραγώνου», που αποτελούν τις κατηγορίες των μεθόδων.

Το πλήκτρο «εύρεσης ριζών» εμφανίζει την βοήθεια για τις μεθόδους εύρεσης ριζών.

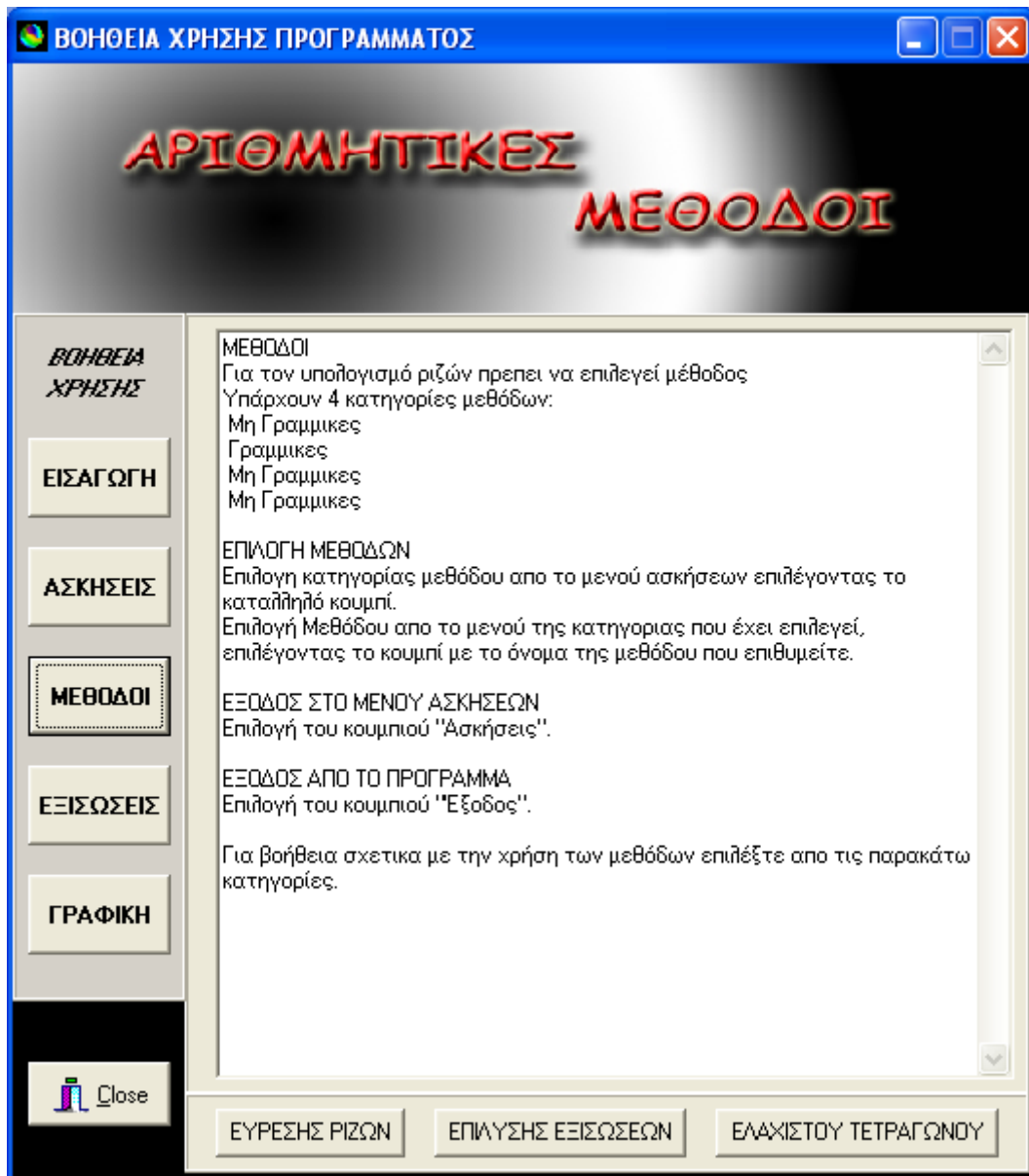
Το πλήκτρο «επίλυση εξισώσεων» εμφανίζει την βοήθεια για τις μεθόδους επίλυσης συστημάτων.

Το πλήκτρο «ελαχίστου τετραγώνου» εμφανίζει την βοήθεια για τη μέθοδο ελαχίστων τετραγώνων.

Το πλήκτρο «εξισώσεις» εμφανίζει την βοήθεια για τη φόρμα συναρτήσεων.

Το πλήκτρο «γραφική» εμφανίζει την βοήθεια για τις γραφικές στις μεθόδους εύρεσης ριζών.

Το πλήκτρο «Close» κλείνει τη φόρμα.



Εικόνα 2.2.16

Τα *meto* που χρησιμοποιούνται είναι μόνο για ανάγνωση και δεν μπορούν να αλλάξουν από το χρήστη.

## ΦΟΡΜΑ ΒΟΗΘΕΙΑΣ ΜΕΘΟΔΩΝ

Η φόρμα βοήθειας μεθόδων του προγράμματος εμφανίζεται επιλέγοντας το πλήκτρο «βοήθεια μεθόδου» από το μενού **Βοήθεια**.

Η φόρμα υλοποιείται από 6 Bit Buttons, 3 Memo και 1 PageControl με 3 TabSheet (Εικ. 2.2.17, 2.2.18, 2.2.19).

Τα 5 πλήκτρα από τα 6 εμφανίζουν τη βοήθεια για κάθε μέθοδο. Αναλυτικότερα, τα πλήκτρα φορτώνουν αρχεία .txt (text) που περιέχουν την βοήθεια στα memo.

Το πλήκτρο «Close» κλείνει την φόρμα.

Το Page Control αποτελείται από τρεις σελίδες (TabSheet)

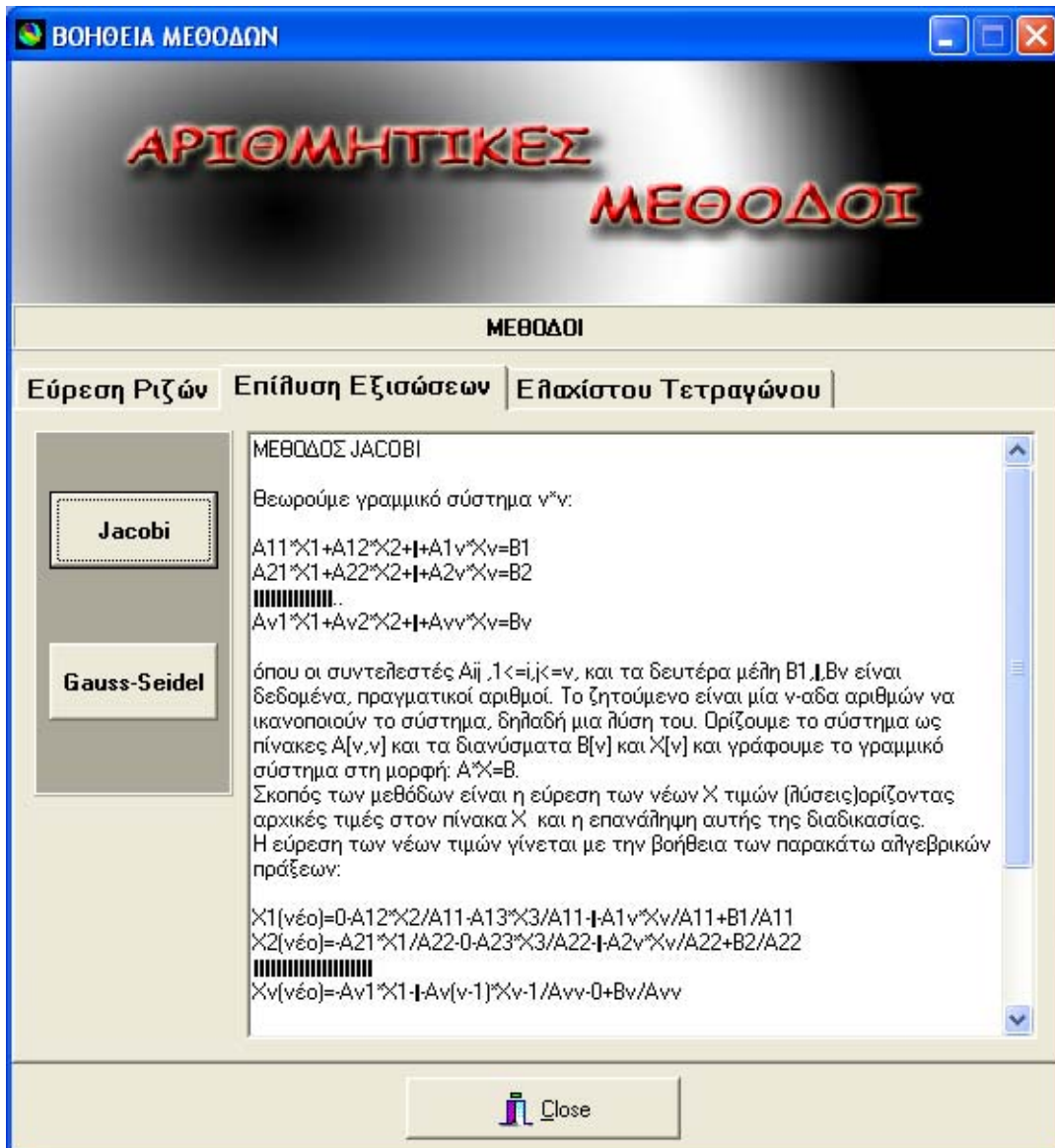
- Τη σελίδα «Εύρεσης ριζών», που περιέχει 3 Bit Buttons και ένα memo (διχοτόμηση, τέμνουσά/regula falsi και Newton) που εμφανίζουν τη βοήθεια για τις αντίστοιχες μεθόδους (Εικ. 2.2.17).



Εικόνα 2.2.17

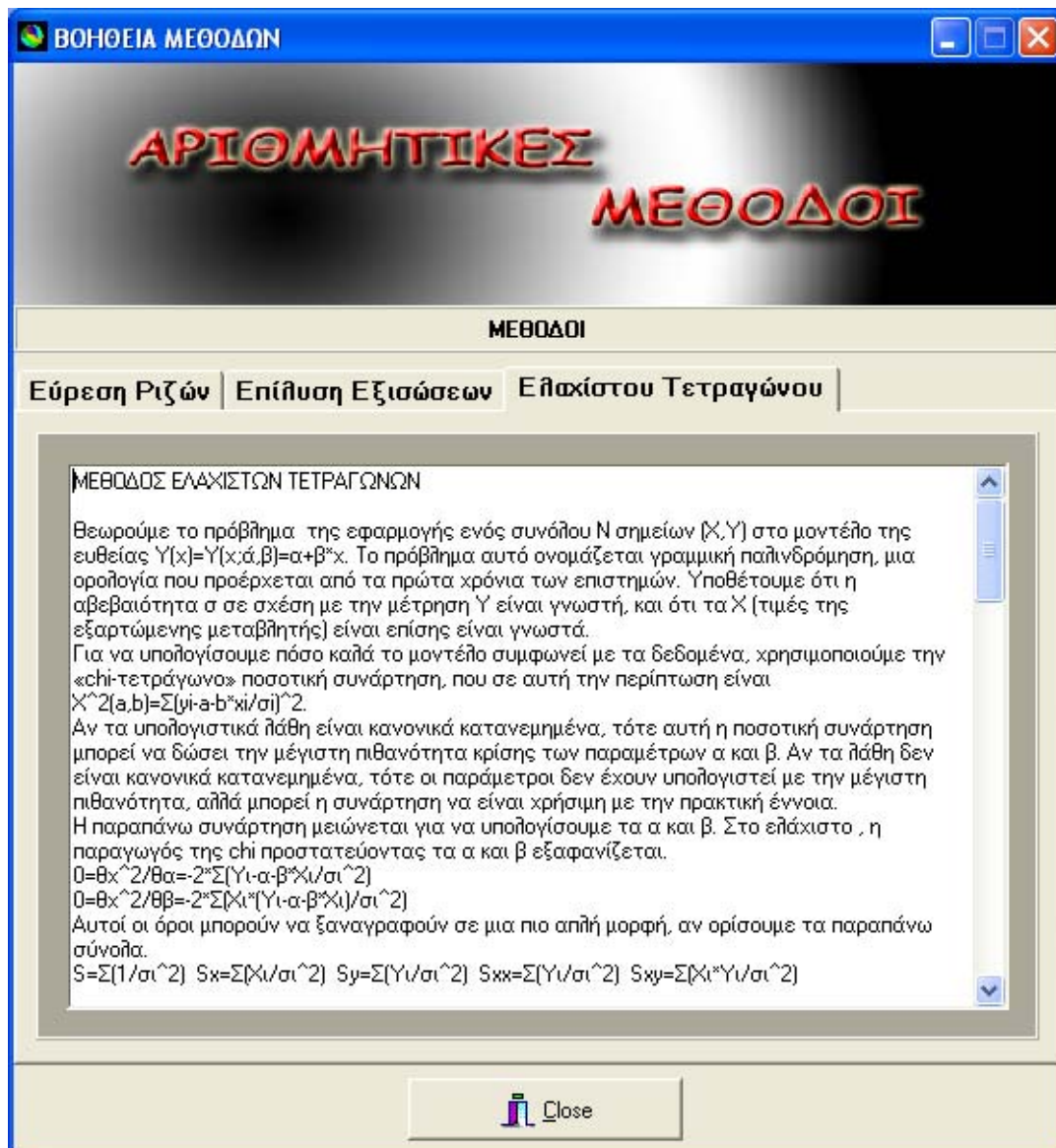


- Τη σελίδα «επίλυση εξισώσεων» που περιέχει 2 Bit Buttons και ένα memo («Jacobi», «Gauss-Seidel») που εμφανίζουν τη βοήθεια για τις αντίστοιχες μεθόδους (Εικ. 2.2.18).



Εικόνα 2.2.18

- Τη σελίδα «ελαχίστου τετραγώνου» που περιέχει ένα memo και εμφανίζει την βοήθεια για τη μέθοδο ελαχίστων τετραγώνων (Εικ.2.2.19)



Εικόνα 2.2.19

Τα memo που χρησιμοποιούνται είναι μόνο για ανάγνωση και δεν μπορούν να αλλάξουν από το χρήστη.

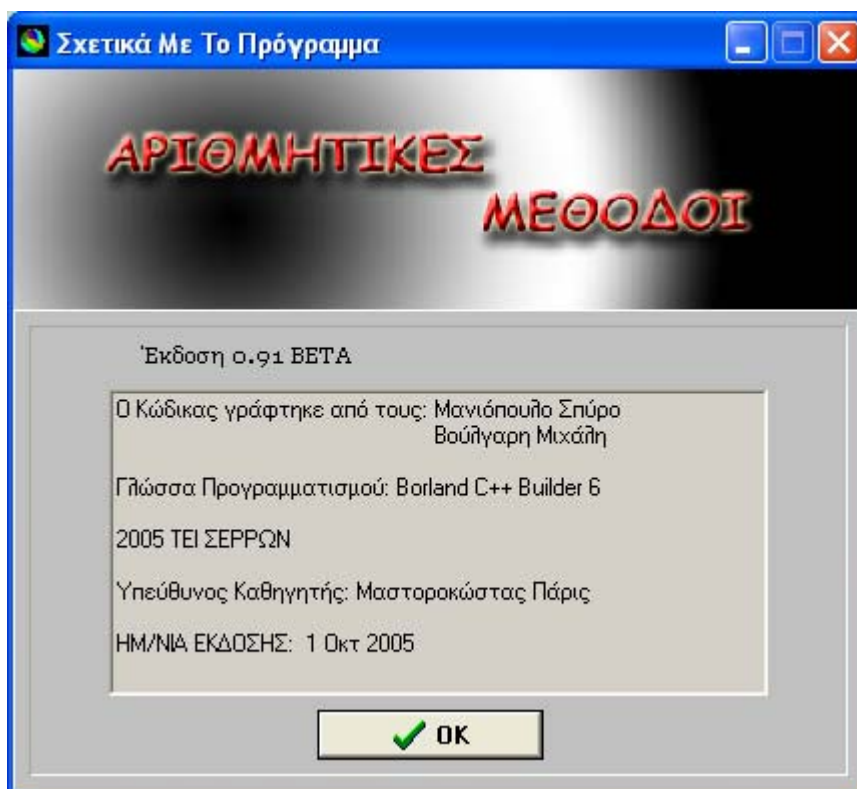
## ΦΟΡΜΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟ ΠΡΟΓΡΑΜΜΑ

Η φόρμα σχετικά με το πρόγραμμα εμφανίζεται επιλέγοντας το πλήκτρο «σχετικά με» από το μενού Βοήθεια.

Η φόρμα υλοποιείται από 1 Bit Button, 1 memo, 1 Label και μία εικόνα - επικεφαλίδα της φόρμας (Εικ. 2.2.20).

Το πλήκτρο «OK» κλείνει τη φόρμα.

Το Memo εμφανίζει τα στοιχεία του προγράμματος (δημιουργός, δικαιώματα κ.λ.π.) και είναι μόνο για ανάγνωση.



Εικόνα 2.2.20

## 2.2.2 Γραφικές

Η γραφική παράσταση συνάρτησης παρουσιάζει την ρίζα της συνάρτησης στο πεδίο ΧΥ. Εμφανίζεται αυτόματα έπειτα από την εκτέλεση της μεθόδου.

### ΕΠΙΛΟΓΗ ΕΜΦΑΝΙΣΗΣ ΓΡΑΦΙΚΗΣ

Για να επιλέξετε αν η γραφική θα εμφανίζεται ή όχι επιλέξτε από το μενού Γραφική «*εμφάνιση γραφικής*». Αν η επιλογή είναι τσεκαρισμένη τότε η γραφική εμφανίζεται, αν όχι ,δεν εμφανίζεται.

Εναλλακτικά, για εμφάνιση, επιλέξτε το κουμπί "*εμφάνισε τη γραφική παράσταση*".

### ΕΠΙΛΟΓΕΣ ΓΡΑΦΙΚΗΣ

Η επιλογή χρωμάτων της γραφικής πραγματοποιείται μέσω της φόρμας επιλογές γραφικής. Για να εμφανιστεί η φόρμα επιλέξτε από το μενού Γραφική «*επιλογές γραφικής*». Εδώ μπορείτε να επιλέξετε διάφορα χρώματα για το φόντο της γραφικής, τις τιμές, το σημείο της ρίζας και των αξόνων της γραφικής. Για να επιλέξετε το χρώμα επιλέξτε το κουμπί που υπάρχει δίπλα στην ιδιότητα της οποίας το χρώμα θέλετε να αλλάξετε. Το χρώμα που χρησιμοποιείτε εμφανίζεται σε διπλανό σχήμα.

Αφού επιλέξετε χρώμα πατήστε «OK». Αν δεν επιθυμείτε αλλαγή των χρωμάτων επιλέξτε το κουμπί «Cancel».

### **2.2.3 Συναρτήσεις**

Οι επιπλέον συναρτήσεις που χρησιμοποιούνται στον πρόγραμμα είναι τρεις.

Η συνάρτηση **IsNumber()** που ελέγχει αν το στοιχείο που εισήγαγε ο χρήστης στο Edit είναι αριθμός ή όχι.

Η συνάρτηση **GramFunc()** που υπολογίζει τέσσερα είδη εξισώσεων:

- Ευθείας
- Τετραγώνου
- Λογαρίθμου
- Κλάσματος

ανάλογα με ότι έχει επιλέξει ο χρήστης.

Η συνάρτηση **Func()** που υπολογίζει την εξίσωση ευθείας.

Η συνάρτηση **DerivFunc()** που υπολογίζει τις παραγωγούς των συναρτήσεων της **GramFunc()** (εκτός της εξίσωσης ευθείας).

### **2.3 Χρήση του προγράμματος**

Το πρόγραμμα αυτό σχεδιάστηκε για τα λειτουργικά συστήματα Microsoft Windows 98/XP και για συστήματα AMD & INTEL.

#### **ΠΡΟΣΟΧΗ!**

Εισαγωγή πολύ υψηλών τιμών στα πεδία στις μεθόδους μπορεί να οδηγήσει σε μη εκτέλεση της μεθόδου και σε λανθασμένες γραφικές παραστάσεις. Αυτό το φαινόμενο εμφανίζεται λόγω περιορισμών των μεταβλητών.

# Κεφάλαιο 3ο

**ΠΑΡΟΥΣΙΑΣΗ ΤΟΥ ΚΩΔΙΚΑ**

### **3.1 Περιγραφή στοιχείων κώδικα**

Ο κώδικας του προγράμματος αποτελείται από 16 τμήματα συνολικού μεγέθους περίπου 8,5 χιλιάδων γραμμών.

Τα 16 τμήματα του κώδικά είναι:

- ✓ [Unit 1](#): Περιλαμβάνει την εισαγωγική φόρμα και τον κώδικα για την εισαγωγή στο πρόγραμμα.
- ✓ [Unit 2](#): Περιλαμβάνει τη φόρμα επιλογής μεθόδου εύρεσης ρίζας και τον κώδικα για την επιλογή μεθόδου.
- ✓ [Unit 3](#): Περιλαμβάνει τη φόρμα της μεθόδου διχοτόμησης και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 4](#): Περιλαμβάνει τη φόρμα της μεθόδου regula falsi και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 5](#): Περιλαμβάνει τη φόρμα σχετικά με το πρόγραμμα.
- ✓ [Unit 6](#): Περιλαμβάνει τη φόρμα της μεθόδου τέμνουσας και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 7](#): Περιλαμβάνει τη φόρμα της μεθόδου του Newton και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 8](#): Περιλαμβάνει τη φόρμα της αποδοτικότητας μεθόδου και τον κώδικα για τον υπολογισμό της μεθόδου.
- ✓ [Unit 9](#): Περιλαμβάνει τη φόρμα ασκήσεων και τον κώδικα για την επιλογή άσκησης.
- ✓ [Unit 10](#): Περιλαμβάνει τη φόρμα συναρτήσεων και τον κώδικα για την επιλογή εξισώσεων.
- ✓ [Unit 11](#): Περιλαμβάνει τη φόρμα επιλογής μεθόδου επίλυσης συστημάτων και τον κώδικα για την επιλογή μεθόδου.
- ✓ [Unit 12](#): Περιλαμβάνει τη φόρμα της μεθόδου Jacobi και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 13](#): Περιλαμβάνει τη φόρμα της μεθόδου Gauss - Seidel και τον κώδικα για την εκτέλεση της μεθόδου.
- ✓ [Unit 14](#): Περιλαμβάνει τη φόρμα της βοήθειας χρήσης και τον κώδικα για την επιλογή βοήθειας.

- ✓ [Unit 15](#): Περιλαμβάνει τη φόρμα της βοήθειας μεθόδων και τον κώδικα για την επιλογή βοήθειας.
- ✓ [Unit 16](#): Περιλαμβάνει τη φόρμα της μεθόδου ελαχίστων τετράγωνων και τον κώδικα για την εκτέλεση της μεθόδου.

Στον κώδικά επίσης χρησιμοποιούνται τρεις συνάρτησης που δεν είναι μέλη της γλώσσας προγραμματισμού και σχεδιάστηκαν από την αρχή.

Οι συναρτήσεις είναι:

Η συνάρτηση **IsNumber()** που είναι μορφής **BOOL** (λογικής) και δίνει σωστό ή λάθος σαν αποτέλεσμα. Δέχεται χαρακτήρες πεδίων Edit box.

```
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-' && edt[i]!='.')
    itis=false;
  }
  return itis;
}
```

Η συνάρτηση **Gramfunc()** που είναι μορφής **double** και δίνει σαν έξοδο πραγματικούς αριθμούς. Δέχεται μία πραγματική τιμή σαν είσοδο.

```
double gramfunc(double x)
{
  double y,a,b,c;
  a=EXISOSEIS->Edit1->Text.ToDouble();
  b=EXISOSEIS->Edit2->Text.ToDouble();
  if(EXISOSEIS->Label1->Visible)
    y=a*x+b;
  if(EXISOSEIS->Label2->Visible)
  {
    c=EXISOSEIS->Edit3->Text.ToDouble();
    y=a*x*x+b*x+c;
  }
}
```



```

if(EXISOSEIS->Label3->Visible)
    y=a*log(x)+b;
if(EXISOSEIS->Label4->Visible)
    y=a*1/x+b;
return y;
}

```

Η συνάρτηση **func()** που είναι μορφής **double** και δίνει σαν έξοδο πραγματικούς αριθμούς. Δέχεται τρεις πραγματικές τιμές σαν είσοδο.

```

double func(double x)
{
    double y,a,b;
    a=Form16->Edit2->Text.ToDouble();
    b=Form16->Edit3->Text.ToDouble();
    y=a+b*x;
    return y;
}

```

Η συνάρτηση **Derivfunc()** που είναι μορφής **double** και δίνει σαν έξοδο πραγματικούς αριθμούς. Δέχεται μία πραγματική τιμή σαν είσοδο.

```

double derivfunc(double x)
{
    double a,b,y;
    a=EXISOSEIS->Edit1->Text.ToDouble();
    b=EXISOSEIS->Edit2->Text.ToDouble();
    if(EXISOSEIS->Label2->Visible)
        y=2*a*x+b;
    if(EXISOSEIS->Label3->Visible)
    {
        if(x==0)
            MessageBox(NULL,"Μεγαλύτερο του 0","Error",MB_OK);
        else
            y=a*1/x;
    }
    if(EXISOSEIS->Label4->Visible)
        y=-a*1/(x*x);
    return y;
}

```

## 3.2 Κώδικας

Στις παρακάτω σελίδες ακολουθεί πλήρης ο κώδικας του προγράμματος με σχόλια.

### Unit 1 (Εισαγωγική φόρμα)

```
//-----  
//Program: Numerical Recipes  
//Version: 1.00  
//Programed By: Maniopoylos Spyros  
//              Voulgaris Mixalis  
//-----  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
#include "Unit2.h"  
#include "Unit9.h"  
//-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
void __fastcall TForm1::Timer1Timer(TObject *Sender)  
{  
    Form1->Hide();  
}  
//-----  
void __fastcall TForm1::FormHide(TObject *Sender)  
{  
    ASKHSEIS->Show();  
}
```

## Unit 2 (Φόρμα επιλογής μεθόδου εύρεσης ρίζας)

```
#include<stdio.h>
#include <vcl.h>
#pragma hdrstop
#include "Unit2.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit9.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
//-----
__fastcall TForm2::TForm2(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm2::BitBtn5Click(TObject *Sender)
{
    Form2->Hide();
    ASKHSEIS->Show();
}
//-----
void __fastcall TForm2::BitBtn1Click(TObject *Sender)
{
    Form2->Hide();
    Form3->Show();
}
//-----
void __fastcall TForm2::BitBtn3Click(TObject *Sender)
{
    Form2->Hide();
    Form4->Show();
}
//-----
void __fastcall TForm2::BitBtn2Click(TObject *Sender)
{
    Form2->Hide();
    Form6->Show();
}
//-----
void __fastcall TForm2::BitBtn4Click(TObject *Sender)
{
    Form2->Hide();
    Form7->Show();
}
//-----
void __fastcall TForm2::BitBtn6Click(TObject *Sender)
{
    Application->Terminate();
}
```

## Unit 3 (Μέθοδος διχοτόμησης)

```
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include <math.hpp>
#include <stdlib.h>
#include "Unit3.h"
#include "Unit2.h"
#include "Unit5.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
TColor LineColor=clYellow;
TColor ImageCanBrush=clNavy;
TColor ImageCanPen=clMaroon;
//-----
__fastcall TForm3::TForm3(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
double gramfunc(double x)
{
    double y,a,b,c;
    a=EXISOSEIS->Edit1->Text.ToDouble();
    b=EXISOSEIS->Edit2->Text.ToDouble();
    if(EXISOSEIS->Label1->Visible)
        y=a*x+b;
    if(EXISOSEIS->Label2->Visible)
        {c=EXISOSEIS->Edit3->Text.ToDouble();
        y=a*x*x+b*x+c;
        }
    if(EXISOSEIS->Label3->Visible)
        {
            y=a*log(x)+b;
        }
    if(EXISOSEIS->Label4->Visible)
        y=a*1/x+b;
    return y;
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!=',' && edt[i]!='-' && edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm3::N13Click(TObject *Sender)
{
  Application->Terminate();
}
//-----
void __fastcall TForm3::N5Click(TObject *Sender)
{
  if(N5->Checked)
  {
    Label17->Visible=false;
    Shape2->Visible=false;
    Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
    Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
    Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
    Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
    Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
    Label18->Visible=false; Label19->Visible=false; Label20->Visible=false;
    Label21->Visible=false; Label22->Visible=false; Label23->Visible=false;
  }
}
```

```

Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

    N5->Checked=false;
    Imagen->Visible=false;
    Shape1->Visible=false;
}
else
{
Label17->Visible=true;
Shape2->Visible=true;
Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;
Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

    N5->Checked=true;
    Imagen->Visible=true;
    Shape1->Visible=true;
}
}
//-----
void __fastcall TForm3::N8Click(TObject *Sender)
{
Form3->Hide();
Form2->Show();
}
//-----
void __fastcall TForm3::N12Click(TObject *Sender)
{
AboutForm->ShowModal();
}
//-----
void __fastcall TForm3::N7Click(TObject *Sender)
{
EfficiencyForm->Show();
}
//-----
void __fastcall TForm3::Button1Click(TObject *Sender)
{
BOOL FuncEnabled=false, AllTextFull=false;
if(Edit6->Text!="" && Edit7->Text!="" && Edit10->Text!="" &&
Edit11->Text!="")
AllTextFull=true;
if(EXISOSEIS->StaticText1->Visible)
FuncEnabled=true;
else
MessageBox(NULL,"Δεν Υπάρχει Συνάρτηση!", "Προσοχή!", MB_OK |
MB_ICONSTOP);
if(IsNumber(Edit6->Text) && IsNumber(Edit7->Text)
&& IsNumber(Edit10->Text) && IsNumber(Edit11->Text)
&& AllTextFull && FuncEnabled)
{
Button3->Enabled=true;
Button1->Enabled=false;
Edit8->Text=gramfunc(Edit6->Text.ToDouble());
Edit9->Text=gramfunc(Edit7->Text.ToDouble());
}
else
MessageBox(NULL,"Μη αποδεκτός χαρακτήρας ή άδεια κελιά","Προσοχή!",
MB_OK|MB_ICONSTOP);
}
}
//-----
void __fastcall TForm3::Button2Click(TObject *Sender)
{
Edit1->Clear();
Edit2->Clear();
Edit3->Clear();
Edit4->Clear();
Edit5->Clear();
}

```

```

Edit6->Clear();
Edit7->Clear();
Edit8->Clear();
Edit9->Clear();
Edit10->Clear();
Edit11->Clear();
CheckBox1->Checked=false;
CheckBox2->Checked=false;
RadioButton3->Checked=false;
RadioButton4->Checked=false;
}
//-----
void __fastcall TForm3::FormClose(TObject *Sender, TCloseAction &Action)
{
Application->Terminate();
}
//-----
void __fastcall TForm3::N2Click(TObject *Sender)
{
EXISOSEIS->Show();
}
//-----
void __fastcall TForm3::Button3Click(TObject *Sender)
{double a,b,x,Limit,lastx;
int NumOfRep,i,Counter=0;
a=Edit6->Text.ToDouble();
b=Edit7->Text.ToDouble();
NumOfRep=Edit10->Text.ToInt();
Limit=Edit11->Text.ToDouble();

if(RadioButton4->Checked==true)
{
if((gramfunc(a)*gramfunc(b))>=0)
{MessageBox(NULL,"Η Συνθήκη πρέπει να ισχύει" , "Σφάλμα!" ,
MB_OK|MB_ICONSTOP);
}
else
{x=a;
do
{lastx=x;
x=(a+b)/2;
if(Counter==0)
Edit3->Text=x;
if((gramfunc(a)*gramfunc(x))<0)
b=x;
else
if((gramfunc(a)*gramfunc(x))>0)
a=x;
Counter++;
}while(Counter<NumOfRep && fabs(lastx-x)>=Limit);

if(Counter==NumOfRep)
{MessageBox(NULL,"Η Μέθοδος Δε Συγκλίνει" , "Αποτελέσματα Μεθόδου" ,
MB_OK|MB_ICONINFORMATION);
Edit2->Text=Counter;
}
else
{
MessageBox(NULL,"Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
MB_OK|MB_ICONINFORMATION);
Edit1->Text=x;
Edit2->Text=Counter;
Edit4->Text=a;
Edit5->Text=b;
Button4->Click();
}
}
Button3->Enabled=false;

MessageBox(NULL,"Πραγματοποίηση Επίλυσης Κατευθείαν" , "Υπόδειξη" ,
MB_OK);
}
else if(RadioButton3->Checked==true)
{
if((gramfunc(a)*gramfunc(b))>=0)
{MessageBox(NULL,"Η Συνθήκη πρέπει να ισχύει" , "Σφάλμα!" , MB_OK |
MB_ICONSTOP);
}
}
}

```

```

}
else
{Button3->Enabled=false;
lastx=a;
x=(a+b)/2;
Edit3->Text=x;
Edit1->Text=x;
if((gramfunc(a)*gramfunc(x))<0)
{Edit4->Text=a;
Edit5->Text=x;
}
else
if((gramfunc(a)*gramfunc(x))>0)
{
Edit4->Text=x;
Edit5->Text=b;
}
Counter++;
Edit2->Text=Counter;
Button4->Click();
if(Counter>=NumOfRep || fabs(lastx-x)<Limit)
{
if(Counter==NumOfRep)
{MessageBox(NULL,"Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
MB_OK|MB_ICONINFORMATION);
}
else
{
MessageBox(NULL,"Η Μέθοδος Συγκλίνει!", "Αποτελέσματα Μεθόδου ",
MB_OK|MB_ICONINFORMATION);
Button4->Click();
}
MessageBox(NULL,"Πραγματοποίηση Επίλυσης Βήμα - Βήμα" , "Υπόδειξη",
MB_OK);
}
else
{Button6->Enabled=true;
Button7->Enabled=true;
}
}
}
else if(RadioButton3->Checked==false && RadioButton4->Checked==false)
MessageBox(NULL," Τρόπος Επίλυσης;;", "Επιλογές",MB_OK);
}
//-----
void __fastcall TForm3::CheckBox2Click(TObject *Sender)
{
if(CheckBox2->Checked==true)
{
Label15->Enabled=true;
Edit11->Enabled=true;
Edit11->SetFocus();
}

else
{
Label15->Enabled=false;
Edit11->Enabled=false;
}
}
//-----
void __fastcall TForm3::Button4Click(TObject *Sender)
{
int Height,Width,X0,Y0,xg,yg;
double xgr,ygr,x1gr,y1gr,x2gr,y2gr,XScale,YScale,xstep;
double x,Ed6,Ed7,Ed8,Ed9;

if(Edit1->Text=="")
{
MessageBox(NULL,"Δεν υπάρχουν δεδομένα" , "Προσοχή" , MB_OK |
MB_ICONINFORMATION);
Label18->Visible=false;Label19->Visible=false;Label20->Visible=false;
Label21->Visible=false;Label22->Visible=false;Label23->Visible=false;
Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;
}
}

```

```

Label17->Visible=false;
Shape2->Visible=false;
Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
}

else
{
Ed6=Edit6->Text.ToDouble();
Ed7=Edit7->Text.ToDouble();
Ed8=Edit8->Text.ToDouble();
Ed9=Edit9->Text.ToDouble();

Image1->Visible=true;
Shape1->Visible=true;

Label17->Visible=true;
Shape2->Visible=true;
Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

Label26->Caption=Max(fabs(Ed6), fabs(Ed7));
Label18->Caption=(Label26->Caption)*(-1.0);
Label23->Caption=Label26->Caption.ToDouble()/4.0;
Label24->Caption=Label26->Caption.ToDouble()/2.0;
Label25->Caption=(Label26->Caption.ToDouble()/4.0)*3.0;
Label19->Caption=(Label25->Caption)*(-1.0);
Label20->Caption=(Label24->Caption)*(-1.0);
Label21->Caption=(Label23->Caption)*(-1.0);

Label27->Caption=Max(fabs(Ed8), fabs(Ed9));
Label32->Caption=(Label27->Caption)*(-1.0);
Label28->Caption=(Label27->Caption.ToDouble()/3.0)*2.0;
Label29->Caption=(Label27->Caption.ToDouble())/3.0;
Label30->Caption=(Label29->Caption.ToDouble())*(-1.0);
Label31->Caption=(Label28->Caption.ToDouble())*(-1.0);

if(Edit6->Text<Edit7->Text)
{
if(Edit1->Text=="")
MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
MB_ICONINFORMATION);
else
{
xgr=Edit1->Text.ToDouble();

Height=Image1->Picture->Height;
Width=Image1->Picture->Width;

X0=Width/2;
Y0=Height/2;

Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect(Rect(0,0,Width,Height));

Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;
}
}
}

```



```

x1gr=Edit6->Text.ToDouble();
x2gr=Edit7->Text.ToDouble();
y1gr=Edit8->Text.ToDouble();
y2gr=Edit9->Text.ToDouble();

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=gramfunc(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=gramfunc(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg,yg);
}
}

else if(Edit6->Text>Edit7->Text)
{
if(Edit1->Text=="")
MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
MB_ICONINFORMATION);
else
{
xgr=Edit1->Text.ToDouble();

Height=Image1->Picture->Height;
Width=Image1->Picture->Width;

X0=Width/2;
Y0=Height/2;

Image1->Canvas->Brush->Color=ImageCanBrush;
Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect(Rect(0,0,Width,Height));

Image1->Canvas->Pen->Color=ImageCanPen;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Edit7->Text.ToDouble();
x2gr=Edit6->Text.ToDouble();
y1gr=Edit9->Text.ToDouble();
y2gr=Edit8->Text.ToDouble();

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=gramfunc(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=gramfunc(xgr);

```

```

        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }

}
} //megalo else
if (Edit1->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα" , "Προσοχή" , MB_OK |
                MB_ICONINFORMATION);
else
{
    if (Edit1->Text>0)
    {
        Label17->Caption=Edit1->Text;
        Shape2->Top=((Image1->Height)/2) + (Image1->Top))-4;
        Shape2->Left=Image1->Left + (Image1->Width)/2 + ((Label17->
        >Caption.ToDouble())*48)*(3.9/(Label26->Caption.ToDouble()))-1;
    }
    else
    {
        Label17->Caption=Edit1->Text;
        Shape2->Top=((Image1->Height)/2) + (Image1->Top))-4;
        Shape2->Left=Image1->Left + (Image1->Width)/2 + ((Label17->
        >Caption.ToDouble())*48)*(4.1/(Label26->Caption.ToDouble()))-1;
    }
}

Label27->Width=17;
Label28->Width=17;
Label29->Width=17;
Label30->Width=20;
Label31->Width=20;
Label32->Width=20;
Label118->Width=20;
Label119->Width=20;
Label20->Width=20;
Label21->Width=20;
Label22->Width=18;
Label23->Width=17;
Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
}
//-----
void __fastcall TForm3::FormCreate(TObject *Sender)
{
    int Height=Image1->Picture->Height;
    int Width=Image1->Picture->Width;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Label17->Caption=="  --";
    Label18->Caption=="---";Label19->Caption=="---";Label20->Caption=="  --";
    Label21->Caption=="---";Label22->Caption=="---";Label23->Caption=="  --";
    Label24->Caption=="---";Label25->Caption=="---";Label26->Caption=="  --";
    Label27->Caption=="---";Label28->Caption=="---";Label29->Caption=="  --";
    Label30->Caption=="---";Label31->Caption=="---";Label32->Caption=="  --";
}
//-----
void __fastcall TForm3::Edit6Change(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm3::FormHide(TObject *Sender)
{
    Button2->Click();
    Label14->Enabled=false;
    Edit10->Enabled=false;
    Label15->Enabled=false;
    Edit11->Enabled=false;
}

```

```

}
//-----
void __fastcall TForm3::CheckBox1Click(TObject *Sender)
{
    if(CheckBox1->Checked==true)
    {
        Labell14->Enabled=true;
        Edit10->Enabled=true;
        Edit10->SetFocus();
    }
    else
    {
        Labell14->Enabled=false;
        Edit10->Enabled=false;
    }
}
//-----
void __fastcall TForm3::RadioButton3Click(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
}
//-----
void __fastcall TForm3::RadioButton4Click(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm3::Button6Click(TObject *Sender)
{
    double a,b,x,Limit,lastx;
    int NumOfRep,Counter;
    Counter=Edit2->Text.ToInt();
    NumOfRep=Edit10->Text.ToInt();
    Limit=Edit11->Text.ToDouble();
    a=Edit4->Text.ToDouble();
    b=Edit5->Text.ToDouble();
    lastx=Edit1->Text.ToDouble();
    x=(a+b)/2;
    Edit1->Text=x;
    if((gramfunc(a)*gramfunc(x))<0)
    {Edit4->Text=a;
      Edit5->Text=x;
    }
    else
    if((gramfunc(a)*gramfunc(x))>0)
    {
        Edit4->Text=x;
        Edit5->Text=b;
    }
    Counter++;
    Edit2->Text=Counter;
    Button4->Click();
    if(Counter>=NumOfRep || fabs(lastx-x)<Limit)
    {
        if(Counter==NumOfRep)
        {MessageBox(NULL,"Η Μέθοδος Δεν Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
                    MB_OK|MB_ICONINFORMATION);
        }
        else
        {
            MessageBox(NULL,"Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
                        MB_OK|MB_ICONINFORMATION);
        }
        MessageBox(NULL,"Πραγματοποίηση Επίλυσης Βήμα-Βήμα" , "Υπόδειξη" ,
                    MB_OK);
        Button6->Enabled=false;
        Button7->Enabled=false;
    }
}
//-----
void __fastcall TForm3::Button7Click(TObject *Sender)
{
    MessageBox(NULL,"Η Επίλυση Βήμα-Βήμα Διακόπηκε!" , "ΜΕΘΟΔΟΣ" ,

```

```

        MB_OK | MB_ICONSTOP);
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm3::N14Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape2->Brush->Color = ColorDialog1->Color;
        Label17->Font->Color = ColorDialog1->Color;
    }
}
//-----
void __fastcall TForm3::N15Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape3->Brush->Color = ColorDialog1->Color;
        Shape4->Brush->Color = ColorDialog1->Color;
        Shape5->Brush->Color = ColorDialog1->Color;
        Shape6->Brush->Color = ColorDialog1->Color;
        Shape7->Brush->Color = ColorDialog1->Color;
        Shape8->Brush->Color = ColorDialog1->Color;
        Shape9->Brush->Color = ColorDialog1->Color;
        Shape10->Brush->Color = ColorDialog1->Color;
        Shape11->Brush->Color = ColorDialog1->Color;
        Shape12->Brush->Color = ColorDialog1->Color;
        Shape13->Brush->Color = ColorDialog1->Color;
        Shape14->Brush->Color = ColorDialog1->Color;
        Shape15->Brush->Color = ColorDialog1->Color;
        Shape16->Brush->Color = ColorDialog1->Color;
        Shape17->Brush->Color = ColorDialog1->Color;

        Label18->Font->Color = ColorDialog1->Color;
        Label19->Font->Color = ColorDialog1->Color;
        Label20->Font->Color = ColorDialog1->Color;
        Label21->Font->Color = ColorDialog1->Color;
        Label22->Font->Color = ColorDialog1->Color;
        Label23->Font->Color = ColorDialog1->Color;
        Label24->Font->Color = ColorDialog1->Color;
        Label25->Font->Color = ColorDialog1->Color;
        Label26->Font->Color = ColorDialog1->Color;
        Label27->Font->Color = ColorDialog1->Color;
        Label28->Font->Color = ColorDialog1->Color;
        Label29->Font->Color = ColorDialog1->Color;
        Label30->Font->Color = ColorDialog1->Color;
        Label31->Font->Color = ColorDialog1->Color;
        Label32->Font->Color = ColorDialog1->Color;

        Label27->Width=17;
        Label28->Width=17;
        Label29->Width=17;
        Label30->Width=20;
        Label31->Width=20;
        Label32->Width=20;
        Label18->Width=20;
        Label19->Width=20;
        Label20->Width=20;
        Label21->Width=20;
        Label22->Width=18;
        Label23->Width=17;
        Label24->Width=17;
        Label25->Width=17;
        Label26->Width=17;
    }
}
//-----
void __fastcall TForm3::N16Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanBrush = ColorDialog1->Color;
    Button4->Click();
}
//-----
void __fastcall TForm3::N17Click(TObject *Sender)
{

```

```

    if (ColorDialog1->Execute())
        LineColor = ColorDialog1->Color;
        Button4->Click();
}
//-----
void __fastcall TForm3::N18Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanPen = ColorDialog1->Color;
        Button4->Click();
}
//-----
void __fastcall TForm3::N10Click(TObject *Sender)
{
    MethHelpForm->Show();
    MethHelpForm->TabSheet1->Show();
    MethHelpForm->BitBtn1->Click();
}
//-----
void __fastcall TForm3::N11Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn3->Click();
    ProgHelpForm->Button1->Click();
}

```

## Unit 4 (Μέθοδος regula falsi)

```
#include <vcl.h>
#pragma hdrstop
#include<math.h>
#include<math.hpp>
#include<stdlib.h>
#include "Unit4.h"
#include "Unit5.h"
#include "Unit2.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
TColor LineColor=clYellow;
TColor ImageCanBrush=clNavy;
TColor ImageCanPen=clMaroon;
//-----
__fastcall TForm4::TForm4(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
double gramfunc(double x)
{
    double y,a,b,c;
    a=EXISOSEIS->Edit1->Text.ToDouble();
    b=EXISOSEIS->Edit2->Text.ToDouble();
    if(EXISOSEIS->Label1->Visible)
        y=a*x+b;
    if(EXISOSEIS->Label2->Visible)
    {
        c=EXISOSEIS->Edit3->Text.ToDouble();
        y=a*x*x+b*x+c;
    }
    if(EXISOSEIS->Label3->Visible)
    {
        if(Form4->Edit6->Text <= 0 || Form4->Edit7->Text <= 0)
        {
            MessageBox(NULL,"Μεγαλύτερο του 0","Error",MB_OK);
            Form4->Button3->Enabled=false;
        }
        else
        {
            y=a*log(x)+b;
            Form4->Button3->Enabled=true;
        }
    }
    if(EXISOSEIS->Label4->Visible)
        y=a*1/x+b;
    return y;
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-' && edt[i]!='+')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm4::N5Click(TObject *Sender)
{
    if (N5->Checked)
    {
        Label17->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
    }
}
```

```

Shape9->Visible=false;Shape10->Visible=false;Shape11->Visible=false;
Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;

Label18->Visible=false;Label19->Visible=false;Label20->Visible=false;
Label21->Visible=false;Label22->Visible=false;Label23->Visible=false;
Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

    N5->Checked=false;
    Image1->Visible=false;
    Shape1->Visible=false;
}
else
{
Label17->Visible=true;
Shape2->Visible=true;
Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

    N5->Checked=true;
    Image1->Visible=true;
    Shape1->Visible=true;
}
}
//-----
void __fastcall TForm4::N8Click(TObject *Sender)
{
Form4->Hide();
Form2->Show();
}
//-----
void __fastcall TForm4::N12Click(TObject *Sender)
{
AboutForm->ShowModal();
}
//-----
void __fastcall TForm4::N13Click(TObject *Sender)
{
Application->Terminate();
}
//-----
void __fastcall TForm4::RadioButton1Click(TObject *Sender)
{
Button1->Enabled=true;
Button3->Enabled=false;
}
//-----
void __fastcall TForm4::RadioButton2Click(TObject *Sender)
{
Button1->Enabled=true;
Button3->Enabled=false;
Button6->Enabled=false;
Button7->Enabled=false;
}
//-----
void __fastcall TForm4::N7Click(TObject *Sender)
{
EfficiencyForm->Show();
}
//-----
void __fastcall TForm4::FormClose(TObject *Sender, TCloseAction &Action)
{
Application->Terminate();
}
//-----
void __fastcall TForm4::Button2Click(TObject *Sender)

```

```

{
    Edit1->Clear();
    Edit2->Clear();
    Edit3->Clear();
    Edit4->Clear();
    Edit5->Clear();
    Edit6->Clear();
    Edit7->Clear();
    Edit8->Clear();
    Edit9->Clear();
    Edit10->Clear();
    Edit11->Clear();
    CheckBox1->Checked=false;
    CheckBox2->Checked=false;
    RadioButton1->Checked=false;
    RadioButton2->Checked=false;
}
//-----
void __fastcall TForm4::Button1Click(TObject *Sender)
{
    BOOL FuncEnabled=false, AllTextFull=false;
    if (Edit6->Text!="" && Edit7->Text!="" && Edit10->Text!=""
        && Edit11->Text!="")
        AllTextFull=true;
    if (EXISOSEIS->StaticText1->Visible)
        FuncEnabled=true;
    else
        MessageBox(NULL, "Δεν Υπάρχει Συνάρτηση!", "Προσοχή!", MB_OK |
            MB_ICONSTOP);
    if (IsNumber(Edit6->Text) && IsNumber(Edit7->Text) &&
        IsNumber(Edit10->Text) && IsNumber(Edit11->Text) && AllTextFull
        && FuncEnabled)
    {
        Button3->Enabled=true;
        Button1->Enabled=false;
        Edit8->Text=gramfunc(Edit6->Text.ToDouble());
        Edit9->Text=gramfunc(Edit7->Text.ToDouble());
    }
    else
        MessageBox(NULL, "Μη αποδεκτός χαρακτήρας ή άδεια κελιά", "Προσοχή!",
            MB_OK|MB_ICONSTOP);
}
//-----
void __fastcall TForm4::Button3Click(TObject *Sender)
{
    double a,b,x,Limit,lastx;
    int NumOfRep,i,Counter=0;
    a=Edit6->Text.ToDouble();
    b=Edit7->Text.ToDouble();
    NumOfRep=Edit10->Text.ToInt();
    Limit=Edit11->Text.ToDouble();

    if (RadioButton2->Checked==true)
    {
        if ((gramfunc(a)*gramfunc(b))>=0)
            {MessageBox(NULL, "Η Συνθήκη πρέπει να ισχύει", "Σφάλμα!", MB_OK |
                MB_ICONSTOP);
            }
    }
    else
    {x=a;
        do
        {lastx=x;
            x=b-((b-a)/(gramfunc(b)-gramfunc(a)))*gramfunc(b);
            if (Counter==0)
                Edit3->Text=x;
            if ((gramfunc(a)*gramfunc(x))<0)
                b=x;
            else
                if ((gramfunc(a)*gramfunc(x))>0)
                    a=x;
            Counter++;
        }while (Counter<NumOfRep && fabs(lastx-x)>=Limit);
        if (Counter==NumOfRep)
            {MessageBox(NULL, "Η Μέθοδος Δεν Συγκλίνει!", "Αποτελέσματα Μεθόδου",
                MB_OK | MB_ICONINFORMATION);
            Edit1->Text=x;
            Edit2->Text=Counter;
        }
    }
}

```



```

    Edit4->Text=a;
    Edit5->Text=b;
}
else
{
    MessageBox(NULL,"Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου",
        MB_OK|MB_ICONINFORMATION);
    Edit1->Text=x;
    Edit2->Text=Counter;
    Edit4->Text=a;
    Edit5->Text=b;
    Button4->Click();
    Button4->Click();
}
}
Button3->Enabled=false;
MessageBox(NULL,"Πραγματοποίηση Επίλυσης Κατευθείαν" , "Υπόδειξη" ,
    MB_OK);
}
else if(RadioButton1->Checked==true)
{
    if((gramfunc(a)*gramfunc(b))>=0)
    {MessageBox(NULL,"Η Συνθήκη πρέπει να ισχύει" , "Σφάλμα!" , MB_OK |
        MB_ICONSTOP);
    }
    else
    {Button3->Enabled=false;
    lastx=a;
    x=b-((b-a)/(gramfunc(b)-gramfunc(a)))*gramfunc(b);
    Edit3->Text=x;
    Edit1->Text=x;
    if((gramfunc(a)*gramfunc(x))<0)
    {Edit4->Text=a;
    Edit5->Text=x;
    }
    else
    if((gramfunc(a)*gramfunc(x))>0)
    {
        Edit4->Text=x;
        Edit5->Text=b;
    }
    Counter++;
    Edit2->Text=Counter;
    Button4->Click();
    Button4->Click();
    if(Counter>=NumOfRep || fabs(lastx-x)<Limit)
    {
        if(Counter==NumOfRep)
        {MessageBox(NULL,"Η Μέθοδος Δεν Συγκλίνει!" ,"Αποτελέσματα Μεθόδου",
            MB_OK|MB_ICONINFORMATION);
        }
        else
        {
            MessageBox(NULL,"Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
                MB_OK|MB_ICONINFORMATION);
            }MessageBox(NULL,"Πραγματοποίηση Επίλυσης Βήμα-Βήμα" , "Υπόδειξη" ,
                MB_OK);
        }
        else
        {Button6->Enabled=true;
        Button7->Enabled=true;
        }
    }
}
else if(RadioButton1->Checked==false && RadioButton2->Checked==false)
MessageBox(NULL,"Τρόπος Επίλυσης;;;" , "Επιλογές",MB_OK);
}
//-----
void __fastcall TForm4::N2Click(TObject *Sender)
{
    EXISOSEIS->Show();
}
//-----
void __fastcall TForm4::Edit6Change(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
}

```

```

Button6->Enabled=false;
Button7->Enabled=false;
}
//-----
void __fastcall TForm4::FormHide(TObject *Sender)
{
    Button2->Click();
    Label14->Enabled=false;
    Edit10->Enabled=false;
    Label15->Enabled=false;
    Edit11->Enabled=false;
}
//-----
void __fastcall TForm4::CheckBox1Click(TObject *Sender)
{
    if (CheckBox1->Checked==true)
    {
        Label14->Enabled=true;
        Edit10->Enabled=true;
        Edit10->SetFocus();
    }
    else
    {
        Label14->Enabled=false;
        Edit10->Enabled=false;
    }
}
//-----
void __fastcall TForm4::CheckBox2Click(TObject *Sender)
{
    if (CheckBox2->Checked==true)
    {
        Label15->Enabled=true;
        Edit11->Enabled=true;
        Edit11->SetFocus();
    }
    else
    {
        Label15->Enabled=false;
        Edit11->Enabled=false;
    }
}
//-----
void __fastcall TForm4::Button4Click(TObject *Sender)
{
    int Height,Width,X0,Y0,xg,yg;
    double xgr,ygr,xlgr,ylgr,x2gr,y2gr,XScale,YScale,xstep;
    double x,Ed6,Ed7,Ed8,Ed9;

    if (Edit1->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
            MB_ICONINFORMATION);

        Label18->Visible=false;Label19->Visible=false;Label20->Visible=false;
        Label21->Visible=false;Label22->Visible=false;Label23->Visible=false;
        Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
        Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
        Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

        Label17->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
        Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }
    else
    {
        Ed6=Edit6->Text.ToDouble();
        Ed7=Edit7->Text.ToDouble();
        Ed8=Edit8->Text.ToDouble();
        Ed9=Edit9->Text.ToDouble();
    }
}

```

```

Image1->Visible=true;
Shape1->Visible=true;

Label17->Visible=true;
Shape2->Visible=true;
Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

Label26->Caption=Max (fabs (Ed6) , fabs (Ed7)) ;
Label18->Caption=(Label26->Caption)*(-1.0);
Label23->Caption=Label26->Caption.ToDouble()/4.0;
Label24->Caption=Label26->Caption.ToDouble()/2.0;
Label25->Caption=(Label26->Caption.ToDouble()/4.0)*3.0;
Label19->Caption=(Label25->Caption)*(-1.0);
Label20->Caption=(Label24->Caption)*(-1.0);
Label21->Caption=(Label23->Caption)*(-1.0);

Label27->Caption=Max (fabs (Ed8) , fabs (Ed9)) ;
Label32->Caption=(Label27->Caption)*(-1.0);
Label28->Caption=(Label27->Caption.ToDouble()/3.0)*2.0;
Label29->Caption=(Label27->Caption.ToDouble()/3.0);
Label30->Caption=(Label29->Caption.ToDouble())*(-1.0);
Label31->Caption=(Label28->Caption.ToDouble())*(-1.0);

if (Edit6->Text<Edit7->Text)
{
    if (Edit1->Text=="")
        MessageBox (NULL,"Δεν υπάρχουν δεδομένα" , "Προσοχή" , MB_OK |
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit1->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect (Rect (0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo (X0,0);
        Image1->Canvas->LineTo (X0,Height);
        Image1->Canvas->MoveTo (0,Y0);
        Image1->Canvas->LineTo (Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Edit6->Text.ToDouble();
        x2gr=Edit7->Text.ToDouble();
        y1gr=Edit8->Text.ToDouble();
        y2gr=Edit9->Text.ToDouble();

        xstep=(x2gr-x1gr)/100.0;

        XScale=Width/2/Max (fabs (x1gr) , fabs (x2gr)) ;
        YScale=Height/2/Max (fabs (y1gr) , fabs (y2gr)) ;
    } //else

    ygr=gramfunc (x1gr) ;
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo (xg, yg) ;

    for (xgr=x1gr; xgr<=x2gr; xgr+=xstep)

```

```

        {
            ygr=gramfunc(xgr);
            xg=xgr*XScale+X0;
            yg=-ygr*YScale+Y0;
            Image1->Canvas->LineTo(xg,yg);
        }
    }

else if(Edit6->Text>Edit7->Text)
{
    if(Edit1->Text=="")
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα" , "Προσοχή" , MB_OK |
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit1->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Edit7->Text.ToDouble();
        x2gr=Edit6->Text.ToDouble();
        y1gr=Edit9->Text.ToDouble();
        y2gr=Edit8->Text.ToDouble();

        xstep=(x2gr-x1gr)/100.0;

        XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
        YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
    }//else

    ygr=gramfunc(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=gramfunc(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

} //megalos else
if(Edit1->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα" , "Προσοχή" , MB_OK |
        MB_ICONINFORMATION);
else
{
    if(Edit1->Text>0)
    {
        Label17->Caption=Edit1->Text;
        Shape2->Top=((Image1->Height)/2) + (Image1->Top)-4;
        Shape2->Left=Image1->Left+(Image1->Width)/2+((Label17->
            Caption.ToDouble())*48)*(3.9/(Label26->Caption.ToDouble()))-1;
    }
    else
    {
        Label17->Caption=Edit1->Text;
        Shape2->Top=((Image1->Height)/2) + (Image1->Top)-4;
        Shape2->Left=Image1->Left+(Image1->Width)/2+((Label17->

```

```

        Caption.ToDouble() * 48) * (4.1 / (Label26->Caption.ToDouble())) - 1;
    }
}

Label27->Width=17;
Label28->Width=17;
Label29->Width=17;
Label30->Width=20;
Label31->Width=20;
Label32->Width=20;
Label18->Width=20;
Label19->Width=20;
Label20->Width=20;
Label21->Width=20;
Label22->Width=18;
Label23->Width=17;
Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
}
//-----
void __fastcall TForm4::Button7Click(TObject *Sender)
{
    MessageBox(NULL, "Η Επίλυση Βήμα-Βήμα Διακόπηκε!" , "ΜΕΘΟΔΟΣ" , MB_OK |
        MB_ICONSTOP);
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm4::Button6Click(TObject *Sender)
{
    double a,b,x,Limit,lastx;
    int NumOfRep,Counter;
    Counter=Edit2->Text.ToInt();
    NumOfRep=Edit10->Text.ToInt();
    Limit=Edit11->Text.ToDouble();
    a=Edit4->Text.ToDouble();
    b=Edit5->Text.ToDouble();
    lastx=Edit1->Text.ToDouble();
    x=b - ((b-a) / (gramfunc(b) - gramfunc(a))) * gramfunc(b);
    Edit1->Text=x;
    if ((gramfunc(a) * gramfunc(x)) < 0)
    {
        Edit4->Text=a;
        Edit5->Text=x;
    }
    else
    if ((gramfunc(a) * gramfunc(x)) > 0)
    {
        Edit4->Text=x;
        Edit5->Text=b;
    }
    Counter++;
    Edit2->Text=Counter;
    Button4->Click();
    if (Counter >= NumOfRep || fabs(lastx-x) < Limit)
    {
        if (Counter == NumOfRep)
        {
            MessageBox(NULL, "Η Μέθοδος Δεν Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
                MB_OK | MB_ICONINFORMATION);
        }
        else
        {
            MessageBox(NULL, "Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου" ,
                MB_OK | MB_ICONINFORMATION);
        }
        MessageBox(NULL, "Πραγματοποίηση Επίλυσης Βήμα-Βήμα" , "Υπόδειξη" ,
            MB_OK);
        Button6->Enabled=false;
        Button7->Enabled=false;
    }
}
//-----
void __fastcall TForm4::FormCreate(TObject *Sender)
{
    Label17->Caption=="  --";
    Label18->Caption=="--";Label19->Caption=="--";Label20->Caption==" --";
    Label21->Caption=="--";Label22->Caption=="--";Label23->Caption==" --";
}

```

```

Label24->Caption=="--";Label25->Caption=="--";Label26->Caption==" --";
Label27->Caption=="--";Label28->Caption=="--";Label29->Caption==" --";
Label30->Caption=="--";Label31->Caption=="--";Label32->Caption==" --";
}
//-----
void __fastcall TForm4::N14Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape2->Brush->Color = ColorDialog1->Color;
        Label17->Font->Color = ColorDialog1->Color;
    }
}
//-----
void __fastcall TForm4::N15Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        LineColor = ColorDialog1->Color;
    Button4->Click();
}
//-----
void __fastcall TForm4::N16Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanPen = ColorDialog1->Color;
    Button4->Click();
}
//-----
void __fastcall TForm4::N17Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanBrush = ColorDialog1->Color;
    Button4->Click();
}
//-----
void __fastcall TForm4::N18Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape3->Brush->Color = ColorDialog1->Color;
        Shape4->Brush->Color = ColorDialog1->Color;
        Shape5->Brush->Color = ColorDialog1->Color;
        Shape6->Brush->Color = ColorDialog1->Color;
        Shape7->Brush->Color = ColorDialog1->Color;
        Shape8->Brush->Color = ColorDialog1->Color;
        Shape9->Brush->Color = ColorDialog1->Color;
        Shape10->Brush->Color = ColorDialog1->Color;
        Shape11->Brush->Color = ColorDialog1->Color;
        Shape12->Brush->Color = ColorDialog1->Color;
        Shape13->Brush->Color = ColorDialog1->Color;
        Shape14->Brush->Color = ColorDialog1->Color;
        Shape15->Brush->Color = ColorDialog1->Color;
        Shape16->Brush->Color = ColorDialog1->Color;
        Shape17->Brush->Color = ColorDialog1->Color;

        Label18->Font->Color = ColorDialog1->Color;
        Label19->Font->Color = ColorDialog1->Color;
        Label20->Font->Color = ColorDialog1->Color;
        Label21->Font->Color = ColorDialog1->Color;
        Label22->Font->Color = ColorDialog1->Color;
        Label23->Font->Color = ColorDialog1->Color;
        Label24->Font->Color = ColorDialog1->Color;
        Label25->Font->Color = ColorDialog1->Color;
        Label26->Font->Color = ColorDialog1->Color;
        Label27->Font->Color = ColorDialog1->Color;
        Label28->Font->Color = ColorDialog1->Color;
        Label29->Font->Color = ColorDialog1->Color;
        Label30->Font->Color = ColorDialog1->Color;
        Label31->Font->Color = ColorDialog1->Color;
        Label32->Font->Color = ColorDialog1->Color;

        Label27->Width=17;
        Label28->Width=17;
        Label29->Width=17;
        Label30->Width=20;
        Label31->Width=20;
        Label32->Width=20;
    }
}

```

```

Label118->Width=20;
Label119->Width=20;
Label120->Width=20;
Label121->Width=20;
Label122->Width=18;
Label123->Width=17;
Label124->Width=17;
Label125->Width=17;
Label126->Width=17;
    }
}
//-----
void __fastcall TForm4::N10Click(TObject *Sender)
{
    MethHelpForm->Show();
    MethHelpForm->TabSheet1->Show();
    MethHelpForm->BitBtn1->Click();
}
//-----
void __fastcall TForm4::N11Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn3->Click();
    ProgHelpForm->Button1->Click();
}

```

## Unit 5 (Σχετικά με το πρόγραμμα)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit5.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TAboutForm *AboutForm;
//-----
__fastcall TAboutForm::TAboutForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TAboutForm::BitBtn1Click(TObject *Sender)
{
    AboutForm->Close();
}
```



## Unit 6 (Μέθοδος τέρνουσας)

```
#include <vcl.h>
#pragma hdrstop
#include <math.h>
#include <math.hpp>
#include <stdlib.h>
#include "Unit6.h"
#include "Unit5.h"
#include "Unit2.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
TColor LineColor=clYellow;
TColor ImageCanBrush=clNavy;
TColor ImageCanPen=clMaroon;
//-----
__fastcall TForm6::TForm6(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
double gramfunc(double x)
{
    double y,a,b,c;
    a=EXISOSEIS->Edit1->Text.ToDouble();
    b=EXISOSEIS->Edit2->Text.ToDouble();
    if(EXISOSEIS->Label1->Visible)
        y=a*x+b;
    if(EXISOSEIS->Label2->Visible)
    {
        c=EXISOSEIS->Edit3->Text.ToDouble();
        y=a*x*x+b*x+c;
    }
    if(EXISOSEIS->Label3->Visible)
        y=a*log(x)+b;
    if(EXISOSEIS->Label4->Visible)
        y=a*1/x+b;
    return y;
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-'&& edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm6::N12Click(TObject *Sender)
{
    AboutForm->ShowModal();
}
//-----
void __fastcall TForm6::N5Click(TObject *Sender)
{
    if(N5->Checked)
    {
        Label17->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
        Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;

        Label18->Visible=false; Label19->Visible=false; Label20->Visible=false;
        Label21->Visible=false; Label22->Visible=false; Label23->Visible=false;
    }
}
```

```

Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

    N5->Checked=false;
    Imagen->Visible=false;
    Shapel->Visible=false;
}
else
{
Label17->Visible=true;
Shape2->Visible=true;
Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

    N5->Checked=true;
    Imagen->Visible=true;
    Shapel->Visible=true;
}
}
//-----
void __fastcall TForm6::N8Click(TObject *Sender)
{
Form6->Hide();
Form2->Show();
}
//-----
void __fastcall TForm6::N13Click(TObject *Sender)
{
Application->Terminate();
}
//-----
void __fastcall TForm6::RadioButton1Click(TObject *Sender)
{
Button1->Enabled=true;
Button3->Enabled=false;
}
//-----
void __fastcall TForm6::RadioButton2Click(TObject *Sender)
{
Button1->Enabled=true;
Button3->Enabled=false;
Button6->Enabled=false;
Button7->Enabled=false;
}
//-----
void __fastcall TForm6::N7Click(TObject *Sender)
{
EfficiencyForm->Show();
}
//-----
void __fastcall TForm6::FormClose(TObject *Sender,
TCloseAction &Action)
{
Application->Terminate();
}
//-----
void __fastcall TForm6::Button2Click(TObject *Sender)
{
Edit1->Clear();
Edit2->Clear();
Edit3->Clear();
Edit6->Clear();
Edit7->Clear();
Edit8->Clear();
Edit9->Clear();
Edit10->Clear();
Edit11->Clear();
}

```

```

CheckBox1->Checked=false;
CheckBox2->Checked=false;
RadioButton1->Checked=false;
RadioButton2->Checked=false;
}
//-----
void __fastcall TForm6::Button1Click(TObject *Sender)
{
    BOOL FuncEnabled=false, AllTextFull=false;
    if (Edit6->Text!="" && Edit7->Text!="" && Edit10->Text!="" &&
        Edit11->Text!="")
        AllTextFull=true;
    if (EXISOSEIS->StaticText1->Visible)
        FuncEnabled=true;
    else
        MessageBox(NULL, "Δεν Υπάρχει Συνάρτηση!", "Προσοχή!", MB_OK |
            MB_ICONSTOP);
    if (IsNumber(Edit6->Text) && IsNumber(Edit7->Text) &&
        IsNumber(Edit10->Text) && IsNumber(Edit11->Text) &&
        AllTextFull && FuncEnabled)
    {
        Button3->Enabled=true;
        Button1->Enabled=false;
        Edit8->Text=gramfunc(Edit6->Text.ToDouble());
        Edit9->Text=gramfunc(Edit7->Text.ToDouble());
    }
    else
        MessageBox(NULL, "Μη αποδεκτός χαρακτήρας ή άδεια κελιά", "Προσοχή!",
            MB_OK|MB_ICONSTOP);
}
//-----
void __fastcall TForm6::Button3Click(TObject *Sender)
{
double a,b,x,Limit,lastx;
int NumOfRep,i,Counter=0;
a=Edit6->Text.ToDouble();
b=Edit7->Text.ToDouble();
NumOfRep=Edit10->Text.ToInt();
Limit=Edit11->Text.ToDouble();

if (RadioButton2->Checked==true)
{
    if ((gramfunc(a)*gramfunc(b))>=0)
    {
        MessageBox(NULL, "Η Συνθήκη πρέπει να ισχύει", "Σφάλμα!", MB_OK | MB_ICONSTOP);
    }
    else
    {
        x=a;
        do
        {
            lastx=x;
            x=b-((b-a)/(gramfunc(b)-gramfunc(a)))*gramfunc(b);
            if (Counter==0)
                Edit3->Text=x;
            a=lastx;
            b=x;
            Counter++;
        }while (Counter<NumOfRep && fabs(lastx-x)>=Limit);
        if (Counter==NumOfRep)
        {
            MessageBox(NULL, "Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
                MB_OK|MB_ICONINFORMATION);
            Edit2->Text=Counter;
        }
        b
        {
            MessageBox(NULL, "Η Μέθοδος Συγκλίνει!", "Αποτελέσματα Μεθόδου",
                MB_OK|MB_ICONINFORMATION);
            Edit1->Text=x;
            Edit2->Text=Counter;
            Button4->Click();
            Button4->Click();
        }
    }
    Button3->Enabled=false;

    MessageBox(NULL, "Πραγματοποίηση Επίλυσης Κατευθείαν", "Υπόδειξη", MB_OK);
}

else if (RadioButton1->Checked==true)
{
    if ((gramfunc(a)*gramfunc(b))>=0)

```

```

{MessageBox(NULL,"Η Συνθήκη πρέπει να ισχύει" , "Σφάλμα!" , MB_OK | MB_ICONSTOP);
}
else
{
  Button3->Enabled=false;
  lastx=a;
  x=b-((b-a)/(gramfunc(b)-gramfunc(a))*gramfunc(b));
  Edit4->Text=lastx;
  Edit3->Text=x;
  Edit1->Text=x;
  Counter++;
  Edit2->Text=Counter;
  Button4->Click();
  Button4->Click();
  if(Counter>=NumOfRep || fabs(lastx-x)<Limit)
  {
    if(Counter==NumOfRep)
    {MessageBox(NULL,"Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
      MB_OK | MB_ICONINFORMATION);
    }
    else
    {
      MessageBox(NULL,"Η Μέθοδος Συγκλίνει!", "Αποτελέσματα Μεθόδου" ,
        MB_OK|MB_ICONINFORMATION);
    }
  }
  MessageBox(NULL,"Προαρμοποίηση Επίλυσης Βήμα - Βήμα","Υπόδειξη", MB_OK);
}
else
{Button6->Enabled=true;
  Button7->Enabled=true;
}
}
}
else if(RadioButton1->Checked==false && RadioButton2->Checked==false)
  MessageBox(NULL,"Τρόπος Επίλυσης;;","Επιλογές",MB_OK);
}
//-----
void __fastcall TForm6::N2Click(TObject *Sender)
{
  EXISOSEIS->Show();
}
//-----

void __fastcall TForm6::Edit6Change(TObject *Sender)
{
  Button1->Enabled=true;
  Button3->Enabled=false;
  Button6->Enabled=false;
  Button7->Enabled=false;
}
//-----
void __fastcall TForm6::FormHide(TObject *Sender)
{
  Button2->Click();
  RadioButton1->Checked=false;
  RadioButton2->Checked=false;
  Label14->Enabled=false;
  Edit10->Enabled=false;
  Label15->Enabled=false;
  Edit11->Enabled=false;
}
//-----
void __fastcall TForm6::CheckBox1Click(TObject *Sender)
{
  if(CheckBox1->Checked==true)
  {
    Label14->Enabled=true;
    Edit10->Enabled=true;
    Edit10->SetFocus();
  }
  else
  {
    Label14->Enabled=false;
    Edit10->Enabled=false;
  }
}
//-----

```

```

void __fastcall TForm6::CheckBox2Click(TObject *Sender)
{
    if(CheckBox2->Checked==true)
    {
        Label15->Enabled=true;
        Edit11->Enabled=true;
        Edit11->SetFocus();
    }

    else
    {
        Label15->Enabled=false;
        Edit11->Enabled=false;
    }
}
//-----
void __fastcall TForm6::Button4Click(TObject *Sender)
{
    int Height,Width,X0,Y0,xg,yg;
    double xgr,ygr,xlgr,ylgr,x2gr,y2gr,XScale,YScale,xstep;
    double x,Ed6,Ed7,Ed8,Ed9;

    if(Edit1->Text=="")
    {
        MessageBox(NULL,"Δεν Υπάρχουν Δεδομένα", "Προσοχή" , MB_OK |
            MB_ICONINFORMATION);

        Label18->Visible=false;Label19->Visible=false;Label20->Visible=false;
        Label21->Visible=false;Label22->Visible=false;Label23->Visible=false;
        Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
        Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
        Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

        Label17->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
        Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }

    else
    {
        Ed6=Edit6->Text.ToDouble();
        Ed7=Edit7->Text.ToDouble();
        Ed8=Edit8->Text.ToDouble();
        Ed9=Edit9->Text.ToDouble();

        Image1->Visible=true;
        Shape1->Visible=true;

        Label17->Visible=true;
        Shape2->Visible=true;
        Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true;Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true;Shape16->Visible=true; Shape17->Visible=true;

        Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
        Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
        Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
        Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
        Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

        Label26->Caption=Max(fabs(Ed6),fabs(Ed7));
        Label18->Caption=(Label26->Caption)*(-1.0);
        Label23->Caption=Label26->Caption.ToDouble()/4.0;
        Label24->Caption=Label26->Caption.ToDouble()/2.0;
        Label25->Caption=(Label26->Caption.ToDouble()/4.0)*3.0;
        Label19->Caption=(Label25->Caption)*(-1.0);
        Label20->Caption=(Label24->Caption)*(-1.0);
        Label21->Caption=(Label23->Caption)*(-1.0);

        Label27->Caption=Max(fabs(Ed8),fabs(Ed9));
        Label32->Caption=(Label27->Caption)*(-1.0);
    }
}

```

```

Label28->Caption=(Label27->Caption.ToDouble()/3.0)*2.0;
Label29->Caption=(Label27->Caption.ToDouble())/3.0;
Label30->Caption=(Label29->Caption.ToDouble()*(-1.0));
Label31->Caption=(Label28->Caption.ToDouble()*(-1.0));

if(Edit6->Text<Edit7->Text)
{
if(Edit1->Text=="")
    MessageBox(NULL,"Δεν Υπάρχουν Δεδομένα","Προσοχή",MB_OK | MB_ICONINFORMATION);
else
{
    xgr=Edit1->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Edit6->Text.ToDouble();
    x2gr=Edit7->Text.ToDouble();
    y1gr=Edit8->Text.ToDouble();
    y2gr=Edit9->Text.ToDouble();

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

    ygr=gramfunc(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=gramfunc(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

else if(Edit6->Text>Edit7->Text)
{
if(Edit1->Text=="")
    MessageBox(NULL,"Δεν Υπάρχουν Δεδομένα","Προσοχή",MB_OK | MB_ICONINFORMATION);
else
{
    xgr=Edit1->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;

```

```

Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Edit7->Text.ToDouble();
x2gr=Edit6->Text.ToDouble();
y1gr=Edit9->Text.ToDouble();
y2gr=Edit8->Text.ToDouble();

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=gramfunc(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=gramfunc(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg,yg);
}
} //megalo else
if(Edit1->Text=="")
MessageBox(NULL,"Δεν Υπάρχουν Δεδομένα" , "Προσοχή" , MB_OK | MB_ICONINFORMATION);
else
{
if(Edit1->Text>0)
{
Label17->Caption=Edit1->Text;
Shape2->Top=((Image1->Height)/2) + (Image1->Top))-4;
Shape2->Left=Image1->Left + (Image1->Width)/2 +
((Label17->Caption.ToDouble())*48)*(3.9/(Label26->Caption.ToDouble()))-1;
}
else
{
Label17->Caption=Edit1->Text;
Shape2->Top=((Image1->Height)/2) + (Image1->Top))-4;
Shape2->Left=Image1->Left + (Image1->Width)/2 +
((Label17->Caption.ToDouble())*48)*(4.1/(Label26->Caption.ToDouble()))-1;
}
}

Label27->Width=17;
Label28->Width=17;
Label29->Width=17;
Label30->Width=20;
Label31->Width=20;
Label32->Width=20;
Label118->Width=20;
Label119->Width=20;
Label20->Width=20;
Label21->Width=20;
Label22->Width=18;
Label23->Width=17;
Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
}
//-----
void __fastcall TForm6::Button7Click(TObject *Sender)
{
MessageBox(NULL,"Η Επίλυση Βήμα - Βήμα Διακόπηκε!" , "ΜΕΘΟΔΟΣ" , MB_OK|MB_ICONSTOP);
Button6->Enabled=false;
Button7->Enabled=false;
}
//-----
void __fastcall TForm6::Button6Click(TObject *Sender)

```

```

{
    double a,b,x,lastx,limit;
    int Counter,NumOfRep;
    limit=Edit11->Text.ToDouble();
    a=Edit4->Text.ToDouble();
    b=Edit1->Text.ToDouble();
    Counter=Edit2->Text.ToInt();
    NumOfRep=Edit10->Text.ToInt();
    lastx=b;
    x=b-((b-a)/(gramfunc(b)-gramfunc(a))*gramfunc(b));
    Edit4->Text=lastx;
    Edit1->Text=x;
    Counter++;
    Edit2->Text=Counter;
    Button4->Click();
    if(Counter>=NumOfRep || fabs(lastx-x)<limit)
    {
        if(Counter==NumOfRep)
            {MessageBox(NULL,"Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
                MB_OK|MB_ICONINFORMATION);
            }
        else
            {
                MessageBox(NULL,"Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου",
                    MB_OK|MB_ICONINFORMATION);
            }
    }
    MessageBox(NULL,"Πραγματοποίηση Επίλυσης Βήμα - Βήμα", "Υπόδειξη" , MB_OK);
    Button6->Enabled=false;
    Button7->Enabled=false;
}
}
//-----
void __fastcall TForm6::N18Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape3->Brush->Color = ColorDialog1->Color;
        Shape4->Brush->Color = ColorDialog1->Color;
        Shape5->Brush->Color = ColorDialog1->Color;
        Shape6->Brush->Color = ColorDialog1->Color;
        Shape7->Brush->Color = ColorDialog1->Color;
        Shape8->Brush->Color = ColorDialog1->Color;
        Shape9->Brush->Color = ColorDialog1->Color;
        Shape10->Brush->Color = ColorDialog1->Color;
        Shape11->Brush->Color = ColorDialog1->Color;
        Shape12->Brush->Color = ColorDialog1->Color;
        Shape13->Brush->Color = ColorDialog1->Color;
        Shape14->Brush->Color = ColorDialog1->Color;
        Shape15->Brush->Color = ColorDialog1->Color;
        Shape16->Brush->Color = ColorDialog1->Color;
        Shape17->Brush->Color = ColorDialog1->Color;

        Label18->Font->Color = ColorDialog1->Color;
        Label19->Font->Color = ColorDialog1->Color;
        Label20->Font->Color = ColorDialog1->Color;
        Label21->Font->Color = ColorDialog1->Color;
        Label22->Font->Color = ColorDialog1->Color;
        Label23->Font->Color = ColorDialog1->Color;
        Label24->Font->Color = ColorDialog1->Color;
        Label25->Font->Color = ColorDialog1->Color;
        Label26->Font->Color = ColorDialog1->Color;
        Label27->Font->Color = ColorDialog1->Color;
        Label28->Font->Color = ColorDialog1->Color;
        Label29->Font->Color = ColorDialog1->Color;
        Label30->Font->Color = ColorDialog1->Color;
        Label31->Font->Color = ColorDialog1->Color;
        Label32->Font->Color = ColorDialog1->Color;

        Label27->Width=17;
        Label28->Width=17;
        Label29->Width=17;
        Label30->Width=20;
        Label31->Width=20;
        Label32->Width=20;
        Label18->Width=20;
        Label19->Width=20;
        Label20->Width=20;
    }
}

```



```

Label21->Width=20;
Label22->Width=18;
Label23->Width=17;
Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
}
}
//-----
void __fastcall TForm6::N17Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanBrush = ColorDialog1->Color;
        Button4->Click();
}
//-----
void __fastcall TForm6::N16Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanPen = ColorDialog1->Color;
        Button4->Click();
}
//-----
void __fastcall TForm6::N15Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        LineColor = ColorDialog1->Color;
        Button4->Click();
}
//-----
void __fastcall TForm6::N14Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        {
            Shape2->Brush->Color = ColorDialog1->Color;
            Label17->Font->Color = ColorDialog1->Color;
        }
}
//-----
void __fastcall TForm6::N10Click(TObject *Sender)
{
    MethHelpForm->Show();
    MethHelpForm->TabSheet1->Show();
    MethHelpForm->BitBtn1->Click();
}
//-----
void __fastcall TForm6::N11Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn3->Click();
    ProgHelpForm->Button1->Click();
}

```

## Unit 7 (Μέθοδος Newton)

```
#include <vcl.h>
#pragma hdrstop
#include<math.h>
#include<math.hpp>
#include<stdlib.h>
#include "Unit7.h"
#include "Unit5.h"
#include "Unit2.h"
#include "Unit8.h"
#include "Unit10.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
TColor LineColor=clYellow;
TColor ImageCanBrush=clNavy;
TColor ImageCanPen=clMaroon;
//-----
__fastcall TForm7::TForm7 (TComponent* Owner)
    : TForm(Owner)
{
}
//-----
BOOL IsNumber (AnsiString edt)
{int i;
  BOOL itis=true;
  for (i=1;i<=edt.Length();i++)
  { if (!isdigit (edt[i]) && edt[i]!=',' && edt[i]!='-' && edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
double gramfunc (double x)
{
  double a,b,c,y;
  a=EXISOSEIS->Edit1->Text.ToDouble();
  b=EXISOSEIS->Edit2->Text.ToDouble();
  if (EXISOSEIS->Label2->Visible)
  {
    c=EXISOSEIS->Edit3->Text.ToDouble();
    y=a*x*x+b*x+c;
  }
  if (EXISOSEIS->Label3->Visible)
  if (Form7->Edit6->Text <= 0)
  {
    MessageBox (NULL, "Μεγαλύτερο του 0", "Error", MB_OK);
    Form7->Button3->Enabled=false;
  }
  else
  {
    y=a*log (x)+b;
    Form7->Button3->Enabled=true;
  }
  if (EXISOSEIS->Label4->Visible)
  y=a*(1/x)+b;
  return y;
}
//-----
double derivfunc (double x)
{
  double a,b,y;
  a=EXISOSEIS->Edit1->Text.ToDouble();
  b=EXISOSEIS->Edit2->Text.ToDouble();
  if (EXISOSEIS->Label2->Visible)
  y=2*a*x+b;
  if (EXISOSEIS->Label3->Visible)
  {
    if (x==0)
      MessageBox (NULL, "Μεγαλύτερο του 0", "Error", MB_OK);
    else
```

```

        y=a*1/x;
    }
    if (EXISOSEIS->Label4->Visible)
        y=-a*1/(x*x);
    return y;
}
//-----
void __fastcall TForm7::N5Click(TObject *Sender)
{
    if (N5->Checked)
    {
        Label117->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
        Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;

        Label118->Visible=false; Label119->Visible=false; Label120->Visible=false;
        Label121->Visible=false; Label122->Visible=false; Label123->Visible=false;
        Label124->Visible=false; Label125->Visible=false; Label126->Visible=false;
        Label127->Visible=false; Label128->Visible=false; Label129->Visible=false;
        Label130->Visible=false; Label131->Visible=false; Label132->Visible=false;

        N5->Checked=false;
        Image1->Visible=false;
        Shape1->Visible=false;
    }
    else
    {
        Label117->Visible=true;
        Shape2->Visible=true;
        Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        Label118->Visible=true; Label119->Visible=true; Label120->Visible=true;
        Label121->Visible=true; Label122->Visible=true; Label123->Visible=true;
        Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
        Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
        Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

        N5->Checked=true;
        Image1->Visible=true;
        Shape1->Visible=true;
    }
}
//-----
void __fastcall TForm7::RadioButton1Click(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
}
//-----
void __fastcall TForm7::RadioButton2Click(TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm7::N12Click(TObject *Sender)
{
    AboutForm->ShowModal();
}
//-----
void __fastcall TForm7::N8Click(TObject *Sender)
{
    Form7->Hide();
    Form2->Show();
}
//-----
void __fastcall TForm7::N13Click(TObject *Sender)

```

```

{
    Application->Terminate();
}
//-----
void __fastcall TForm7::N7Click(TObject *Sender)
{
    EfficiencyForm->Show();
}
//-----
void __fastcall TForm7::FormClose(TObject *Sender,
                                TCloseAction &Action)
{
    Application->Terminate();
}
//-----
void __fastcall TForm7::Button1Click(TObject *Sender)
{
    BOOL FuncEnabled=false,AllTextFull=false,NotZero=true;
    double Ed5, Ed6, lastx;
    int NumOfRep,i,Counter=0;

    if(Edit6->Text!="" && (Edit10->Text!="" && Edit11->Text!=""))
        AllTextFull=true;
    if(EXISOSEIS->StaticText1->Visible)
        FuncEnabled=true;
    else
        MessageBox(NULL,"Δεν Υπάρχει Συνάρτηση!", "Προσοχή!", MB_OK | MB_ICONSTOP);

    if(EXISOSEIS->Label4->Visible && Edit6->Text==0)
        NotZero=false;
    if(IsNumber(Edit6->Text) && IsNumber(Edit10->Text) &&
        IsNumber(Edit11->Text) && AllTextFull && FuncEnabled && NotZero)
    {
        Button3->Enabled=true;
        Button1->Enabled=false;
        lastx=Edit6->Text.ToDouble();
        Edit5->Text=0.1;
        Edit4->Text=gramfunc(lastx);
        Edit8->Text=derivfunc(lastx);
        Edit7->Text=(gramfunc(0.1));
    }
    else
        MessageBox(NULL,"Μη αποδεκτός χαρακτήρας ή άδεια κελιά","Προσοχή!",
        MB_OK|MB_ICONSTOP);
}
//-----
void __fastcall TForm7::Button2Click(TObject *Sender)
{
    Edit1->Clear();
    Edit2->Clear();
    Edit3->Clear();
    Edit6->Clear();
    Edit8->Clear();
    Edit4->Clear();
    Edit10->Clear();
    Edit11->Clear();
}
//-----
void __fastcall TForm7::Button4Click(TObject *Sender)
{
    int randNumber;
    randNumber=random(11);
    Edit6->Text=randNumber;
}
//-----
void __fastcall TForm7::Button3Click(TObject *Sender)
{BOOL ExitByBreak=false;
double x,Limit,lastx;
int NumOfRep,i,Counter=0;
lastx=Edit6->Text.ToDouble();
Edit5->Text=0.1;
Edit4->Text=gramfunc(lastx);
Edit8->Text=derivfunc(lastx);
Edit7->Text=(gramfunc(0.1));
NumOfRep=Edit10->Text.ToInt();
Limit=Edit11->Text.ToDouble();
}

```

```

if (RadioButton2->Checked==true)
{
    do
    {
        if (Counter!=0)
            lastx=x;
        if (derivfunc (lastx)==0)
        {
            ExitByBreak=true;
            break;
        }
        x=lastx-(gramfunc (lastx)/derivfunc (lastx));
        if (Counter==0)
            Edit3->Text=x;
        Counter++;
    }while (Counter<NumOfRep && fabs (lastx-x)>=Limit);
    if (Counter==NumOfRep || ExitByBreak)
{MessageBox (NULL, "Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
    MB_OK|MB_ICONINFORMATION);
    Edit2->Text=Counter;
    }
    else
    {
        MessageBox (NULL, " Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου",
            MB_OK|MB_ICONINFORMATION);
        Edit1->Text=x;
        Edit2->Text=Counter;
        Button5->Click ();
        Button5->Click ();
    }
    Button3->Enabled=false;
    MessageBox (NULL, "Πραγματοποίηση Επίλυσης Κατευθείαν", "Υπόδειξη", MB_OK);
}
if (RadioButton1->Checked==true)
{Button3->Enabled=false;
if (derivfunc (lastx)==0)
{
    ExitByBreak=true;
}
else
{
    x=lastx-(gramfunc (lastx)/derivfunc (lastx));
    Edit3->Text=x;
    Edit1->Text=x;
    Counter++;
    Edit2->Text=Counter;
}
if (Counter>=NumOfRep || fabs (lastx-x)<Limit || ExitByBreak)
{
    if (Counter==NumOfRep || ExitByBreak)
{MessageBox (NULL, "Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
    MB_OK| MB_ICONINFORMATION);
    }
    else
    {
        MessageBox (NULL, "Η Μέθοδος Συγκλίνει!" , "Αποτελέσματα Μεθόδου ",
            MB_OK | MB_ICONINFORMATION);
    }
}
    MessageBox (NULL, "Πραγματοποίηση Επίλυσης Βήμα - Βήμα!", "Υπόδειξη" ,MB_OK);
}
else
{Button6->Enabled=true;
    Button7->Enabled=true;
}
}
else if (RadioButton1->Checked==false && RadioButton2->Checked==false)
    MessageBox (NULL, "Τρόπος Επίλυσης;;", "Επιλογές",MB_OK);
}
//-----
void __fastcall TForm7::Edit6Change (TObject *Sender)
{
    Button1->Enabled=true;
    Button3->Enabled=false;
}
//-----
void __fastcall TForm7::N2Click (TObject *Sender)
{

```

```

EXISOSEIS->Show();
}
//-----
void __fastcall TForm7::FormHide(TObject *Sender)
{
    Button2->Click();
    RadioButton1->Checked=false;
    RadioButton2->Checked=false;
    Label14->Enabled=false;
    Edit10->Enabled=false;
    Label15->Enabled=false;
    Edit11->Enabled=false;
}
//-----
void __fastcall TForm7::Button5Click(TObject *Sender)
{
    int Height, Width, X0,Y0,xg,yg;
    double xgr,ygr,xlgr,ylgr,x2gr,y2gr,XScale,YScale,xstep;
    double x,Ed6,Ed5,Ed4,Ed7;

    if(Edit1->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
            MB_ICONINFORMATION);
        Label18->Visible=false;Label19->Visible=false;Label20->Visible=false;
        Label21->Visible=false;Label22->Visible=false;Label23->Visible=false;
        Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
        Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
        Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

        Label17->Visible=false;
        Shape2->Visible=false;
        Shape3->Visible=false; Shape4->Visible=false; Shape5->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
        Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }

    else
    {
        Ed6=Edit6->Text.ToDouble();
        Ed5 = 0.1;
        Ed4=Edit4->Text.ToDouble();
        Ed7=Edit7->Text.ToDouble();

        Image1->Visible=true;
        Shape1->Visible=true;

        Label17->Visible=true;
        Shape2->Visible=true;
        Shape3->Visible=true; Shape4->Visible=true; Shape5->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        Label18->Visible=true; Label19->Visible=true; Label20->Visible=true;
        Label21->Visible=true; Label22->Visible=true; Label23->Visible=true;
        Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
        Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
        Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

        Label26->Caption=Max(fabs(Ed6), fabs(Ed5));
        Label18->Caption=(Label26->Caption)*(-1.0);
        Label23->Caption=Label26->Caption.ToDouble()/4.0;
        Label24->Caption=Label26->Caption.ToDouble()/2.0;
        Label25->Caption=(Label26->Caption.ToDouble()/4.0)*3.0;
        Label19->Caption=(Label25->Caption)*(-1.0);
        Label20->Caption=(Label24->Caption)*(-1.0);
        Label21->Caption=(Label23->Caption)*(-1.0);

        Label27->Caption=Max(fabs(Ed4), fabs(Ed7));
        Label32->Caption=(Label27->Caption)*(-1.0);
        Label28->Caption=(Label27->Caption.ToDouble()/3.0)*2.0;
        Label29->Caption=(Label27->Caption.ToDouble()/3.0);
        Label30->Caption=(Label29->Caption.ToDouble()*(-1.0));
    }
}

```

```

Label31->Caption=(Label28->Caption.ToDouble())*(-1.0);
if(Edit6->Text<Edit5->Text)
{
if(Edit1->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
                MB_ICONINFORMATION);
else
{
    xgr=Edit1->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Edit6->Text.ToDouble();
    x2gr=0.1;//Edit5->Text.ToDouble();
    y1gr=Edit4->Text.ToDouble();
    y2gr=Edit7->Text.ToDouble();

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

    ygr=gramfunc(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=gramfunc(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

else if(Edit6->Text>Edit5->Text)
{
if(Edit1->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK | MB_ICONINFORMATION);

else
{
    xgr=Edit1->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
}
}

```

```

Image1->Canvas->MoveTo(0, Y0);
Image1->Canvas->LineTo(Width, Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=0.1;
x2gr=Edit6->Text.ToDouble();
y1gr=Edit7->Text.ToDouble();
y2gr=Edit4->Text.ToDouble();

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
} //else

ygr=gramfunc(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg, yg);

for(xgr=x1gr; xgr<=x2gr; xgr+=xstep)
{
ygr=gramfunc(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg, yg);
}
} //megalo else
if(Edit1->Text=="")
MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK | MB_ICONINFORMATION);

else
{
if(Edit1->Text>0)
{
Label17->Caption=Edit1->Text;
Shape2->Top=((Image1->Height)/2) + (Image1->Top)-4;
Shape2->Left=Image1->Left + (Image1->Width)/2 +
((Label17->Caption.ToDouble()*48)*(3.9/(Label26->Caption.ToDouble()))-1);
if(Shape2->Left<=Image1->Left-9 && Shape2->Left>=Image1->Left +
Image1->Width && Shape2->Top<=Image1->Top-9 &&
Shape2->Top<=Image1->Top + Image1->Height)
Shape2->Visible=false;
else Shape2->Visible=true;
}
else
{
Label17->Caption=Edit1->Text;
Shape2->Top=((Image1->Height)/2) + (Image1->Top)-4;
Shape2->Left=Image1->Left+(Image1->Width)/2+
((Label17->Caption.ToDouble()*48)*(4.1/(Label26->Caption.ToDouble()))-1);
if(Shape2->Left<=Image1->Left-9 && Shape2->Left>=Image1->Left+
Image1->Width && Shape2->Top<=Image1->Top-9 &&
Shape2->Top<=Image1->Top+ Image1->Height)
Shape2->Visible=false;
else Shape2->Visible=true;
}
}

Label27->Width=17;
Label28->Width=17;
Label29->Width=17;
Label30->Width=20;
Label31->Width=20;
Label32->Width=20;
Label118->Width=20;
Label119->Width=20;
Label20->Width=20;
Label21->Width=20;
Label22->Width=18;
Label23->Width=17;
Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
}
//-----

```



```

void __fastcall TForm7::FormCreate(TObject *Sender)
{
    int Height=Image1->Picture->Height;
    int Width=Image1->Picture->Width;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));
}
//-----
void __fastcall TForm7::CheckBox1Click(TObject *Sender)
{
    if (CheckBox1->Checked==true)
    {
        Label14->Enabled=true;
        Edit10->Enabled=true;
        Edit10->SetFocus();
    }
    else
    {
        Label14->Enabled=false;
        Edit10->Enabled=false;
    }
}
//-----
void __fastcall TForm7::CheckBox2Click(TObject *Sender)
{
    if (CheckBox2->Checked==true)
    {
        Label15->Enabled=true;
        Edit11->Enabled=true;
        Edit11->SetFocus();
    }

    else
    {
        Label15->Enabled=false;
        Edit11->Enabled=false;
    }
}
//-----
void __fastcall TForm7::Button6Click(TObject *Sender)
{
    BOOL ExitByBreak=false;
    double x,Limit,lastx;
    int NumOfRep,i,Counter;
    lastx=Edit1->Text.ToDouble();
    Counter=Edit2->Text.ToInt();
    Limit=Edit11->Text.ToDouble();
    NumOfRep=Edit10->Text.ToInt();
    if (derivfunc(lastx)==0)
    {
        ExitByBreak=true;
    }
    else
    {
        x=lastx-(gramfunc(lastx)/derivfunc(lastx));
        Edit1->Text=x;
        Counter++;
        Edit2->Text=Counter;
        Button5->Click();
    }
    if (Counter>=NumOfRep || fabs(lastx-x)<Limit || ExitByBreak)
    {
        if (Counter==NumOfRep || ExitByBreak)
        {
            MessageBox(NULL,"Η Μέθοδος Δε Συγκλίνει!", "Αποτελέσματα Μεθόδου",
                MB_OK|MB_ICONINFORMATION);
        }
        else
        {
            MessageBox(NULL," Η Μέθοδος Συγκλίνει! ", "Αποτελέσματα Μεθόδου ",
                MB_OK|MB_ICONINFORMATION);
        }
    }
    MessageBox(NULL,"Πραγματοποίηση Επίλυσης Βήμα-Βήμα!", "Υπόδειξη", MB_OK);
    Button6->Enabled=false;
    Button7->Enabled=false;
}
}

```

```

//-----
void __fastcall TForm7::Button7Click(TObject *Sender)
{
    MessageBox(NULL, "Η Επίλυση Βήμα - Βήμα Διακόπηκε!", "ΜΕΘΟΔΟΣ",
        MB_OK|MB_ICONSTOP);
    Button6->Enabled=false;
    Button7->Enabled=false;
}
//-----
void __fastcall TForm7::N14Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape2->Brush->Color = ColorDialog1->Color;
        Label17->Font->Color = ColorDialog1->Color;
    }
}
//-----
void __fastcall TForm7::N15Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        LineColor = ColorDialog1->Color;
}
//-----
void __fastcall TForm7::N16Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanPen = ColorDialog1->Color;
}
//-----
void __fastcall TForm7::N17Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
        ImageCanBrush = ColorDialog1->Color;
}
//-----
void __fastcall TForm7::N18Click(TObject *Sender)
{
    if (ColorDialog1->Execute())
    {
        Shape3->Brush->Color = ColorDialog1->Color;
        Shape4->Brush->Color = ColorDialog1->Color;
        Shape5->Brush->Color = ColorDialog1->Color;
        Shape6->Brush->Color = ColorDialog1->Color;
        Shape7->Brush->Color = ColorDialog1->Color;
        Shape8->Brush->Color = ColorDialog1->Color;
        Shape9->Brush->Color = ColorDialog1->Color;
        Shape10->Brush->Color = ColorDialog1->Color;
        Shape11->Brush->Color = ColorDialog1->Color;
        Shape12->Brush->Color = ColorDialog1->Color;
        Shape13->Brush->Color = ColorDialog1->Color;
        Shape14->Brush->Color = ColorDialog1->Color;
        Shape15->Brush->Color = ColorDialog1->Color;
        Shape16->Brush->Color = ColorDialog1->Color;
        Shape17->Brush->Color = ColorDialog1->Color;

        Label18->Font->Color = ColorDialog1->Color;
        Label19->Font->Color = ColorDialog1->Color;
        Label20->Font->Color = ColorDialog1->Color;
        Label21->Font->Color = ColorDialog1->Color;
        Label22->Font->Color = ColorDialog1->Color;
        Label23->Font->Color = ColorDialog1->Color;
        Label24->Font->Color = ColorDialog1->Color;
        Label25->Font->Color = ColorDialog1->Color;
        Label26->Font->Color = ColorDialog1->Color;
        Label27->Font->Color = ColorDialog1->Color;
        Label28->Font->Color = ColorDialog1->Color;
        Label29->Font->Color = ColorDialog1->Color;
        Label30->Font->Color = ColorDialog1->Color;
        Label31->Font->Color = ColorDialog1->Color;
        Label32->Font->Color = ColorDialog1->Color;

        Label27->Width=17;
        Label28->Width=17;
        Label29->Width=17;
        Label30->Width=20;
        Label31->Width=20;
    }
}

```

```

Label132->Width=20;
Label118->Width=20;
Label119->Width=20;
Label120->Width=20;
Label121->Width=20;
Label122->Width=18;
Label123->Width=17;
Label124->Width=17;
Label125->Width=17;
Label126->Width=17;
    }
}
//-----
void __fastcall TForm7::N10Click(TObject *Sender)
{
    MethHelpForm->Show();
    MethHelpForm->TabSheet1->Show();
    MethHelpForm->BitBtn1->Click();
}
//-----
void __fastcall TForm7::N11Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn3->Click();
    ProgHelpForm->Button1->Click();
}

```

## Unit 8 (Αποδοτικότητα μεθόδου)

```
#include<math.h>
#include <vcl.h>
#pragma hdrstop
#include "Unit8.h"
#include "Unit3.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit4.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TEfficiencyForm *EfficiencyForm;
//-----
__fastcall TEfficiencyForm::TEfficiencyForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-' && edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TEfficiencyForm::BitBtn1Click(TObject *Sender)
{
  EfficiencyForm->Close();
}
//-----
void __fastcall TEfficiencyForm::Button1Click(TObject *Sender)
{
  double YpolRiza,PragRiza,apotel;
  if(Form3->Visible)
    Edit2->Text=Form3->Edit1->Text;
  else if(Form4->Visible)
    Edit2->Text=Form4->Edit1->Text;
  else if(Form6->Visible)
    Edit2->Text=Form6->Edit1->Text;
  else if(Form7->Visible)
    Edit2->Text=Form7->Edit1->Text;
  if(IsNumber(Edit1->Text) && IsNumber(Edit2->Text) &&
    Edit1->Text!="" && Edit2->Text!="")
  {YpolRiza=Edit2->Text.ToDouble();
    PragRiza=Edit1->Text.ToDouble();
    apotel=100-((fabs(PragRiza-YpolRiza)/fabs(PragRiza))*100);
    Edit3->Text=apotel;
  }
  else
  MessageBox(NULL,"Άδεια Κελιά ή Μη Αποδεκτοί Χαρακτήρες!", "Προσοχή",
    MB_OK|MB_ICONSTOP);
}
//-----
void __fastcall TEfficiencyForm::FormClose(TObject *Sender,
  TCloseAction &Action)
{
  Edit1->Text="";
  Edit2->Text="";
  Edit3->Text="";
}
```

## Unit 9 (Ασκήσεις)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit9.h"
#include "Unit2.h"
#include "Unit11.h"
#include "Unit14.h"
#include "Unit16.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TASKHSEIS *ASKHSEIS;
//-----
__fastcall TASKHSEIS::TASKHSEIS(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TASKHSEIS::BitBtn1Click(TObject *Sender)
{
    Application->Terminate();
}
//-----
void __fastcall TASKHSEIS::BitBtnClick(TObject *Sender)
{
    Form2->Show();
    ASKHSEIS->Hide();
}
//-----
void __fastcall TASKHSEIS::BitBtn2Click(TObject *Sender)
{
    Form11->Show();
    ASKHSEIS->Hide();
}
//-----
void __fastcall TASKHSEIS::BitBtn2Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn2->Click();
}
//-----
void __fastcall TASKHSEIS::BitBtn3Click(TObject *Sender)
{
    Form16->Show();
    ASKHSEIS->Hide();
}
}
```

## Unit 10 (Συναρτήσεις - επιλογή εξισώσεων)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit10.h"
#include "Unit7.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TEXISOSEIS *EXISOSEIS;
//-----
__fastcall TEXISOSEIS::TEXISOSEIS(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if (!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TEXISOSEIS::BitBtn2Click(TObject *Sender)
{
  Timer1->Enabled=false;
  EXISOSEIS->Close();
}
//-----
void __fastcall TEXISOSEIS::Button1Click(TObject *Sender)
{
  int i=ListBox1->ItemIndex;
  if (ListBox1->Items->Strings[i]=="Εξίσωση Ευθείας" &&
      Form7->Visible==false)
  {
    Label1->Show();
    Label7->Visible=false;
    Edit3->Visible=false;
  }
  else if (ListBox1->Items->Strings[i]=="Εξίσωση Ευθείας" &&
           Form7->Visible==true)
  MessageBox(NULL, "Η Εξίσωση Ευθείας Δεν Είναι Αποδεκτή Στη Newton!",
             "Σφάλμα", MB_OK | MB_ICONSTOP);
  if (ListBox1->Items->Strings[i]=="Εξίσωση Τετραγώνου")
  {
    Label2->Show();
    Label7->Visible=true;
    Edit3->Visible=true;
  }
  if (ListBox1->Items->Strings[i]=="Εξίσωση Λογαρίθμου")
  {
    Label3->Show();
    Label7->Visible=false;
    Edit3->Visible=false;
  }
  if (ListBox1->Items->Strings[i]=="Εξίσωση Κλάσματος")
  {
    Label4->Show();
    Label7->Visible=false;
    Edit3->Visible=false;
  }
  Button1->Enabled=false;
}
//-----
void __fastcall TEXISOSEIS::ListBox1Click(TObject *Sender)
{
  Button1->Enabled=true;
  Label1->Hide();
  Label2->Hide();
  Label3->Hide();
  Label4->Hide();
  Label7->Hide();
  Edit3->Hide();
}
```

```

    StaticText1->Visible=false;
    Timer1->Enabled=false;
}
//-----
void __fastcall TEXISOSEIS::BitBtn1Click(TObject *Sender)
{
    BOOL LabelActive=false,EditsFull=false;
    if(Label1->Visible||Label2->Visible || Label3->Visible || Label4->Visible)
        LabelActive=true;
    if(Edit1->Text!="" && Edit2->Text!="")
        EditsFull=true;
    if(LabelActive && EditsFull && IsNumber(Edit1->Text) &&
        IsNumber(Edit2->Text))
        {EXISOSEIS->Hide();
        StaticText1->Visible=true;
        }
    else
    MessageBox(NULL,"Μη Επιλεγµένη Συνάρτηση Η Μη Αποδεκτοί Χαρακτήρες!",
        "Σφάλμα",MB_OK|MB_ICONSTOP);
}
//-----
void __fastcall TEXISOSEIS::FormClose(TObject *Sender,
    TCloseAction &Action)
{int i=ListBox1->ItemIndex;
    Button1->Enabled=false;
    Label1->Visible=false;
    Label2->Visible=false;
    Label3->Visible=false;
    Label4->Visible=false;
    Edit1->Text="";
    Edit2->Text="";
    StaticText1->Visible=false;
    if(ListBox1->Items->Strings[i]=="Εξίσωση Τετραγώνου")
    {Edit3->Text="";
    Label7->Visible=false;
    Edit3->Visible=false;
    }
}
//-----
void __fastcall TEXISOSEIS::Timer1Timer(TObject *Sender)
{
    if(StaticText1->Visible==true)
        StaticText1->Visible=false;
    else StaticText1->Visible=true;
}
//-----
void __fastcall TEXISOSEIS::FormShow(TObject *Sender)
{
    if(StaticText1->Visible==true)
    {
        Timer1->Enabled=true;
        Timer1->Interval=500;
        StaticText1->Width=140;
    }

    else
        Timer1->Enabled=false;
}
//-----
void __fastcall TEXISOSEIS::FormCreate(TObject *Sender)
{
    Button1->Enabled=false;
    Timer1->Enabled=false;
    Label7->Visible=false;
    Edit3->Visible=false;
}
//-----
void __fastcall TEXISOSEIS::Edit1Change(TObject *Sender)
{
    Timer1->Enabled=false;
}
}

```

## Unit 11 (Ψόρμα επιλογής μεθόδου επίλυσης συστημάτων)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit11.h"
#include "Unit2.h"
#include "Unit9.h"
#include "Unit12.h"
#include "Unit13.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm11 *Form11;
//-----
__fastcall TForm11::TForm11(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm11::BitBtn3Click(TObject *Sender)
{
    Form11->Hide();
    ASKHSEIS->Show();
}
//-----
void __fastcall TForm11::BitBtn4Click(TObject *Sender)
{
    Application->Terminate();
}
//-----
void __fastcall TForm11::BitBtn1Click(TObject *Sender)
{
    Form11->Hide();
    Form12->Show();
}
//-----
void __fastcall TForm11::BitBtn2Click(TObject *Sender)
{
    Form11->Hide();
    Form13->Show();
}
```



## Unit 12 (Μέθοδος Jacobi)

```
#include<math.h>
#include <vcl.h>
#pragma hdrstop
#include "Unit12.h"
#include "Unit5.h"
#include "Unit11.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm12 *Form12;
//-----
__fastcall TForm12::TForm12(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if (!isdigit(edt[i]) && edt[i]!=',' && edt[i]!='-'&& edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm12::N4Click(TObject *Sender)
{
  Form12->Close();
}
//-----
void __fastcall TForm12::FormClose(TObject *Sender, TCloseAction &Action)
{
  Application->Terminate();
}
//-----
void __fastcall TForm12::N8Click(TObject *Sender)
{
  AboutForm->ShowModal();
}
//-----
void __fastcall TForm12::N3Click(TObject *Sender)
{
  Form12->Hide();
  Form11->Show();
}
//-----
void __fastcall TForm12::BitBtn1Click(TObject *Sender)
{
  if(IsNumber(Edit1->Text) && IsNumber(Edit2->Text))
  {
    if(Edit1->Text==1)
    {
      A11->Enabled=true; A11->Color=clCream;
      X1->Enabled=true; X1->Color=clCream;
      B1->Enabled=true; B1->Color=clCream;
      BitBtn1->Caption="Ready";
      Label10->Enabled=true;
    }
    else
    if(Edit1->Text==2)
    {
      A11->Enabled=true; A11->Color=clCream;
      A12->Enabled=true; A12->Color=clCream;
      A21->Enabled=true; A21->Color=clCream;
      A22->Enabled=true; A22->Color=clCream;
      X1->Enabled=true; X1->Color=clCream;
      X2->Enabled=true; X2->Color=clCream;
      B1->Enabled=true; B1->Color=clCream;
      B2->Enabled=true; B2->Color=clCream;
    }
  }
}
```

```

    BitBtn1->Caption="Ready";
    Label10->Enabled=true;
}
else
if (Edit1->Text==3)
{
    A11->Enabled=true; A11->Color=clCream;
    A12->Enabled=true; A12->Color=clCream;
    A13->Enabled=true; A13->Color=clCream;
    A21->Enabled=true; A21->Color=clCream;
    A22->Enabled=true; A22->Color=clCream;
    A23->Enabled=true; A23->Color=clCream;
    A31->Enabled=true; A31->Color=clCream;
    A32->Enabled=true; A32->Color=clCream;
    A33->Enabled=true; A33->Color=clCream;
    X1->Enabled=true; X1->Color=clCream;
    X2->Enabled=true; X2->Color=clCream;
    X3->Enabled=true; X3->Color=clCream;
    B1->Enabled=true; B1->Color=clCream;
    B2->Enabled=true; B2->Color=clCream;
    B3->Enabled=true; B3->Color=clCream;
    BitBtn1->Caption="Ready";
    Label10->Enabled=true;
}
else
if (Edit1->Text==4)
{
    A11->Enabled=true; A11->Color=clCream;
    A12->Enabled=true; A12->Color=clCream;
    A13->Enabled=true; A13->Color=clCream;
    A14->Enabled=true; A14->Color=clCream;
    A21->Enabled=true; A21->Color=clCream;
    A22->Enabled=true; A22->Color=clCream;
    A23->Enabled=true; A23->Color=clCream;
    A24->Enabled=true; A24->Color=clCream;
    A31->Enabled=true; A31->Color=clCream;
    A32->Enabled=true; A32->Color=clCream;
    A33->Enabled=true; A33->Color=clCream;
    A34->Enabled=true; A34->Color=clCream;
    A41->Enabled=true; A41->Color=clCream;
    A42->Enabled=true; A42->Color=clCream;
    A43->Enabled=true; A43->Color=clCream;
    A44->Enabled=true; A44->Color=clCream;
    X1->Enabled=true; X1->Color=clCream;
    X2->Enabled=true; X2->Color=clCream;
    X3->Enabled=true; X3->Color=clCream;
    X4->Enabled=true; X4->Color=clCream;
    B1->Enabled=true; B1->Color=clCream;
    B2->Enabled=true; B2->Color=clCream;
    B3->Enabled=true; B3->Color=clCream;
    B4->Enabled=true; B4->Color=clCream;
    BitBtn1->Caption="Ready";
    Label10->Enabled=true;
}
else
if (Edit1->Text==5)
{
    A11->Enabled=true; A11->Color=clCream;
    A12->Enabled=true; A12->Color=clCream;
    A13->Enabled=true; A13->Color=clCream;
    A14->Enabled=true; A14->Color=clCream;
    A15->Enabled=true; A15->Color=clCream;
    A21->Enabled=true; A21->Color=clCream;
    A22->Enabled=true; A22->Color=clCream;
    A23->Enabled=true; A23->Color=clCream;
    A24->Enabled=true; A24->Color=clCream;
    A25->Enabled=true; A25->Color=clCream;
    A31->Enabled=true; A31->Color=clCream;
    A32->Enabled=true; A32->Color=clCream;
    A33->Enabled=true; A33->Color=clCream;
    A34->Enabled=true; A34->Color=clCream;
    A35->Enabled=true; A35->Color=clCream;
    A41->Enabled=true; A41->Color=clCream;
    A42->Enabled=true; A42->Color=clCream;
    A43->Enabled=true; A43->Color=clCream;
    A44->Enabled=true; A44->Color=clCream;
    A45->Enabled=true; A45->Color=clCream;
}

```

```

A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
if (Edit1->Text==6)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;
A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A15->Enabled=true; A15->Color=clCream;
A16->Enabled=true; A16->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A25->Enabled=true; A25->Color=clCream;
A26->Enabled=true; A26->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A35->Enabled=true; A35->Color=clCream;
A36->Enabled=true; A36->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
A45->Enabled=true; A45->Color=clCream;
A46->Enabled=true; A46->Color=clCream;
A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
A56->Enabled=true; A56->Color=clCream;
A61->Enabled=true; A61->Color=clCream;
A62->Enabled=true; A62->Color=clCream;
A63->Enabled=true; A63->Color=clCream;
A64->Enabled=true; A64->Color=clCream;
A65->Enabled=true; A65->Color=clCream;
A66->Enabled=true; A66->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
X6->Enabled=true; X6->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
B6->Enabled=true; B6->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
if (Edit1->Text==7)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;

```

```

A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A15->Enabled=true; A15->Color=clCream;
A16->Enabled=true; A16->Color=clCream;
A17->Enabled=true; A17->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A25->Enabled=true; A25->Color=clCream;
A26->Enabled=true; A26->Color=clCream;
A27->Enabled=true; A27->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A35->Enabled=true; A35->Color=clCream;
A36->Enabled=true; A36->Color=clCream;
A37->Enabled=true; A37->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
A45->Enabled=true; A45->Color=clCream;
A46->Enabled=true; A46->Color=clCream;
A47->Enabled=true; A47->Color=clCream;
A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
A56->Enabled=true; A56->Color=clCream;
A57->Enabled=true; A57->Color=clCream;
A61->Enabled=true; A61->Color=clCream;
A62->Enabled=true; A62->Color=clCream;
A63->Enabled=true; A63->Color=clCream;
A64->Enabled=true; A64->Color=clCream;
A65->Enabled=true; A65->Color=clCream;
A66->Enabled=true; A66->Color=clCream;
A67->Enabled=true; A67->Color=clCream;
A71->Enabled=true; A71->Color=clCream;
A72->Enabled=true; A72->Color=clCream;
A73->Enabled=true; A73->Color=clCream;
A74->Enabled=true; A74->Color=clCream;
A75->Enabled=true; A75->Color=clCream;
A76->Enabled=true; A76->Color=clCream;
A77->Enabled=true; A77->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
X6->Enabled=true; X6->Color=clCream;
X7->Enabled=true; X7->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
B6->Enabled=true; B6->Color=clCream;
B7->Enabled=true; B7->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
    MessageBox(NULL,"Λάθος Δεδομένα!", "Σφάλμα", MB_OK|MB_ICONSTOP);
}
else
    MessageBox(NULL,"Μη αποδεκτός Χαρακτήρας!", "Σφάλμα", MB_OK| MB_ICONSTOP);
}
//-----
void __fastcall TForm12::Edit1Change(TObject *Sender)
{
    Edit2->Text=Edit1->Text;
    Edit4->Text=Edit1->Text;
    BitBtn1->Caption="OK";
}

```

```

Label10->Enabled=false;
//Reset
A11->Enabled=false; A11->Color=clSkyBlue;
A12->Enabled=false; A12->Color=clSkyBlue;
A13->Enabled=false; A13->Color=clSkyBlue;
A14->Enabled=false; A14->Color=clSkyBlue;
A15->Enabled=false; A15->Color=clSkyBlue;
A16->Enabled=false; A16->Color=clSkyBlue;
A17->Enabled=false; A17->Color=clSkyBlue;
A21->Enabled=false; A21->Color=clSkyBlue;
A22->Enabled=false; A22->Color=clSkyBlue;
A23->Enabled=false; A23->Color=clSkyBlue;
A24->Enabled=false; A24->Color=clSkyBlue;
A25->Enabled=false; A25->Color=clSkyBlue;
A26->Enabled=false; A26->Color=clSkyBlue;
A27->Enabled=false; A27->Color=clSkyBlue;
A31->Enabled=false; A31->Color=clSkyBlue;
A32->Enabled=false; A32->Color=clSkyBlue;
A33->Enabled=false; A33->Color=clSkyBlue;
A34->Enabled=false; A34->Color=clSkyBlue;
A35->Enabled=false; A35->Color=clSkyBlue;
A36->Enabled=false; A36->Color=clSkyBlue;
A37->Enabled=false; A37->Color=clSkyBlue;
A41->Enabled=false; A41->Color=clSkyBlue;
A42->Enabled=false; A42->Color=clSkyBlue;
A43->Enabled=false; A43->Color=clSkyBlue;
A44->Enabled=false; A44->Color=clSkyBlue;
A45->Enabled=false; A45->Color=clSkyBlue;
A46->Enabled=false; A46->Color=clSkyBlue;
A47->Enabled=false; A47->Color=clSkyBlue;
A51->Enabled=false; A51->Color=clSkyBlue;
A52->Enabled=false; A52->Color=clSkyBlue;
A53->Enabled=false; A53->Color=clSkyBlue;
A54->Enabled=false; A54->Color=clSkyBlue;
A55->Enabled=false; A55->Color=clSkyBlue;
A56->Enabled=false; A56->Color=clSkyBlue;
A57->Enabled=false; A57->Color=clSkyBlue;
A61->Enabled=false; A61->Color=clSkyBlue;
A62->Enabled=false; A62->Color=clSkyBlue;
A63->Enabled=false; A63->Color=clSkyBlue;
A64->Enabled=false; A64->Color=clSkyBlue;
A65->Enabled=false; A65->Color=clSkyBlue;
A66->Enabled=false; A66->Color=clSkyBlue;
A67->Enabled=false; A67->Color=clSkyBlue;
A71->Enabled=false; A71->Color=clSkyBlue;
A72->Enabled=false; A72->Color=clSkyBlue;
A73->Enabled=false; A73->Color=clSkyBlue;
A74->Enabled=false; A74->Color=clSkyBlue;
A75->Enabled=false; A75->Color=clSkyBlue;
A76->Enabled=false; A76->Color=clSkyBlue;
A77->Enabled=false; A77->Color=clSkyBlue;
X1->Enabled=false; X1->Color=clSkyBlue;
X2->Enabled=false; X2->Color=clSkyBlue;
X3->Enabled=false; X3->Color=clSkyBlue;
X4->Enabled=false; X4->Color=clSkyBlue;
X5->Enabled=false; X5->Color=clSkyBlue;
X6->Enabled=false; X6->Color=clSkyBlue;
X7->Enabled=false; X7->Color=clSkyBlue;
B1->Enabled=false; B1->Color=clSkyBlue;
B2->Enabled=false; B2->Color=clSkyBlue;
B3->Enabled=false; B3->Color=clSkyBlue;
B4->Enabled=false; B4->Color=clSkyBlue;
B5->Enabled=false; B5->Color=clSkyBlue;
B6->Enabled=false; B6->Color=clSkyBlue;
B7->Enabled=false; B7->Color=clSkyBlue;

A11->Text="";
A12->Text="";
A13->Text="";
A14->Text="";
A15->Text="";
A16->Text="";
A17->Text="";
A21->Text="";
A22->Text="";
A23->Text="";
A24->Text="";

```

```

A25->Text="";
A26->Text="";
A27->Text="";
A31->Text="";
A32->Text="";
A33->Text="";
A34->Text="";
A35->Text="";
A36->Text="";
A37->Text="";
A41->Text="";
A42->Text="";
A43->Text="";
A44->Text="";
A45->Text="";
A46->Text="";
A47->Text="";
A51->Text="";
A52->Text="";
A53->Text="";
A54->Text="";
A55->Text="";
A56->Text="";
A57->Text="";
A61->Text="";
A62->Text="";
A63->Text="";
A64->Text="";
A65->Text="";
A66->Text="";
A67->Text="";
A71->Text="";
A72->Text="";
A73->Text="";
A74->Text="";
A75->Text="";
A76->Text="";
A77->Text="";
X1->Text="";
X2->Text="";
X3->Text="";
X4->Text="";
X5->Text="";
X6->Text="";
X7->Text="";
B1->Text="";
B2->Text="";
B3->Text="";
B4->Text="";
B5->Text="";
B6->Text="";
B7->Text="";
}
//-----
void __fastcall TForm12::N2Click(TObject *Sender)
{
    BOOL EditsFull=false;
    int i,j,N=0;
    double A[7][7],X[7],B[7],FX[7],norma=0,limit;
    for (i=0;i<7;i++)
        for (j=0;j<7;j++)
            {A[i][j]=0;
            X[i]=0;
            B[i]=0;
            FX[i]=0;
            }
    if (Label10->Enabled && IsNumber(Edit3->Text) && Edit3->Text!="")
        {N=Edit4->Text.ToInt();
        limit=Edit3->Text.ToDouble();
        if (N==2)
            if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
                IsNumber(A12->Text) && A21->Text!="" && IsNumber(A21->Text) &&
                A22->Text!="" && IsNumber(A22->Text) && X1->Text!="" &&
                IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
                B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
                IsNumber(B2->Text))
                {
                A[0][0]=A11->Text.ToDouble();

```

```

A[0][1]=A12->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if(N==3)
{if(A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A21->Text!="" && IsNumber(A21->Text) && A22->Text!="" &&
IsNumber(A22->Text) && A23->Text!="" && IsNumber(A23->Text) &&
A31->Text!="" && IsNumber(A31->Text) && A32->Text!="" &&
IsNumber(A32->Text) && A33->Text!="" && IsNumber(A33->Text) &&
X1->Text!="" && IsNumber(X1->Text) && X2->Text!="" &&
IsNumber(X2->Text) && X3->Text!="" && IsNumber(X3->Text) &&
B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
IsNumber(B2->Text) && B3->Text!="" && IsNumber(B3->Text))

{ A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if(N==4)
{if(A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A14->Text!="" && IsNumber(A14->Text) && A21->Text!="" &&
IsNumber(A21->Text) && A22->Text!="" && IsNumber(A22->Text) &&
A23->Text!="" && IsNumber(A23->Text) && A24->Text!="" &&
IsNumber(A24->Text) && A31->Text!="" && IsNumber(A31->Text) &&
A32->Text!="" && IsNumber(A32->Text) && A33->Text!="" &&
IsNumber(A33->Text) && A34->Text!="" && IsNumber(A34->Text) &&
A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
A44->Text!="" && IsNumber(A44->Text) && X1->Text!="" &&
IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
X3->Text!="" && IsNumber(X3->Text) && X4->Text!="" &&
IsNumber(X4->Text) && B1->Text!="" && IsNumber(B1->Text) &&
B2->Text!="" && IsNumber(B2->Text) && B3->Text!="" &&
IsNumber(B3->Text) && B4->Text!="" && IsNumber(B4->Text))
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
A[2][3]=A34->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();

```

```

A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
A[3][3]=A44->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if(N==5)
{if(A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A14->Text!="" && IsNumber(A14->Text) && A15->Text!="" &&
IsNumber(A15->Text) && A21->Text!="" && IsNumber(A21->Text) &&
A22->Text!="" && IsNumber(A22->Text) && A23->Text!="" &&
IsNumber(A23->Text) && A24->Text!="" && IsNumber(A24->Text) &&
A25->Text!="" && IsNumber(A25->Text) && A31->Text!="" &&
IsNumber(A31->Text) && A32->Text!="" && IsNumber(A32->Text) &&
A33->Text!="" && IsNumber(A33->Text) && A34->Text!="" &&
IsNumber(A34->Text) && A35->Text!="" && IsNumber(A35->Text) &&
A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
A44->Text!="" && IsNumber(A44->Text) && A45->Text!="" &&
IsNumber(A45->Text) && A51->Text!="" && IsNumber(A51->Text) &&
A52->Text!="" && IsNumber(A52->Text) && A53->Text!="" &&
IsNumber(A53->Text) && A54->Text!="" && IsNumber(A54->Text) &&
A55->Text!="" && IsNumber(A55->Text) && X1->Text!="" &&
IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
X3->Text!="" && IsNumber(X3->Text) && X4->Text!="" &&
IsNumber(X4->Text) && X5->Text!="" && IsNumber(X5->Text) &&
B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
IsNumber(B2->Text) && B3->Text!="" && IsNumber(B3->Text) &&
B4->Text!="" && IsNumber(B4->Text) && B5->Text!="" &&
IsNumber(B5->Text))
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[0][4]=A15->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[1][4]=A25->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
A[2][3]=A34->Text.ToDouble();
A[2][4]=A35->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();
A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
A[3][3]=A44->Text.ToDouble();
A[3][4]=A45->Text.ToDouble();
A[4][0]=A51->Text.ToDouble();
A[4][1]=A52->Text.ToDouble();
A[4][2]=A53->Text.ToDouble();
A[4][3]=A54->Text.ToDouble();
A[4][4]=A55->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
X[4]=X5->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();

```



```

    B[4]=B5->Text.ToDouble();
    EditsFull=true;
}
else
MessageBox(NULL,"Ελεειπή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
    MB_OK|MB_ICONSTOP);
}
if (N==6)
{if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
    IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
    A14->Text!="" && IsNumber(A14->Text) && A15->Text!="" &&
    IsNumber(A15->Text) && A16->Text!="" && IsNumber(A16->Text) &&
    A21->Text!="" && IsNumber(A21->Text) && A22->Text!="" &&
    IsNumber(A22->Text) && A23->Text!="" && IsNumber(A23->Text) &&
    A24->Text!="" && IsNumber(A24->Text) && A25->Text!="" &&
    IsNumber(A25->Text) && A26->Text!="" && IsNumber(A26->Text) &&
    A31->Text!="" && IsNumber(A31->Text) && A32->Text!="" &&
    IsNumber(A32->Text) && A33->Text!="" && IsNumber(A33->Text) &&
    A34->Text!="" && IsNumber(A34->Text) && A35->Text!="" &&
    IsNumber(A35->Text) && A36->Text!="" && IsNumber(A36->Text) &&
    A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
    IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
    A44->Text!="" && IsNumber(A44->Text) && A45->Text!="" &&
    IsNumber(A45->Text) && A46->Text!="" && IsNumber(A46->Text) &&
    A51->Text!="" && IsNumber(A51->Text) && A52->Text!="" &&
    IsNumber(A52->Text) && A53->Text!="" && IsNumber(A53->Text) &&
    A54->Text!="" && IsNumber(A54->Text) && A55->Text!="" &&
    IsNumber(A55->Text) && A56->Text!="" && IsNumber(A56->Text) &&
    A61->Text!="" && IsNumber(A61->Text) && A62->Text!="" &&
    IsNumber(A62->Text) && A63->Text!="" && IsNumber(A63->Text) &&
    A64->Text!="" && IsNumber(A64->Text) && A65->Text!="" &&
    IsNumber(A65->Text) && A66->Text!="" && IsNumber(A66->Text) &&
    X1->Text!="" && IsNumber(X1->Text) && X2->Text!="" &&
    IsNumber(X2->Text) && X3->Text!="" && IsNumber(X3->Text) &&
    X4->Text!="" && IsNumber(X4->Text) && X5->Text!="" &&
    IsNumber(X5->Text) && X6->Text!="" && IsNumber(X6->Text) &&
    B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
    IsNumber(B2->Text) && B3->Text!="" && IsNumber(B3->Text) &&
    B4->Text!="" && IsNumber(B4->Text) && B5->Text!="" &&
    IsNumber(B5->Text) && B6->Text!="" && IsNumber(B6->Text))
{
    A[0][0]=A11->Text.ToDouble();
    A[0][1]=A12->Text.ToDouble();
    A[0][2]=A13->Text.ToDouble();
    A[0][3]=A14->Text.ToDouble();
    A[0][4]=A15->Text.ToDouble();
    A[0][5]=A16->Text.ToDouble();
    A[1][0]=A21->Text.ToDouble();
    A[1][1]=A22->Text.ToDouble();
    A[1][2]=A23->Text.ToDouble();
    A[1][3]=A24->Text.ToDouble();
    A[1][4]=A25->Text.ToDouble();
    A[1][5]=A26->Text.ToDouble();
    A[2][0]=A31->Text.ToDouble();
    A[2][1]=A32->Text.ToDouble();
    A[2][2]=A33->Text.ToDouble();
    A[2][3]=A34->Text.ToDouble();
    A[2][4]=A35->Text.ToDouble();
    A[2][5]=A36->Text.ToDouble();
    A[3][0]=A41->Text.ToDouble();
    A[3][1]=A42->Text.ToDouble();
    A[3][2]=A43->Text.ToDouble();
    A[3][3]=A44->Text.ToDouble();
    A[3][4]=A45->Text.ToDouble();
    A[3][5]=A46->Text.ToDouble();
    A[4][0]=A51->Text.ToDouble();
    A[4][1]=A52->Text.ToDouble();
    A[4][2]=A53->Text.ToDouble();
    A[4][3]=A54->Text.ToDouble();
    A[4][4]=A55->Text.ToDouble();
    A[4][5]=A56->Text.ToDouble();
    A[5][0]=A61->Text.ToDouble();
    A[5][1]=A62->Text.ToDouble();
    A[5][2]=A63->Text.ToDouble();
    A[5][3]=A64->Text.ToDouble();
    A[5][4]=A65->Text.ToDouble();
    A[5][5]=A66->Text.ToDouble();
}
}
}

```

```

X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
X[4]=X5->Text.ToDouble();
X[5]=X6->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();
B[4]=B5->Text.ToDouble();
B[5]=B6->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if (N==7)
{if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A14->Text!="" && IsNumber(A14->Text) && A15->Text!="" &&
IsNumber(A15->Text) && A16->Text!="" && IsNumber(A16->Text) &&
A17->Text!="" && IsNumber(A17->Text) && A21->Text!="" &&
IsNumber(A21->Text) && A22->Text!="" && IsNumber(A22->Text) &&
A23->Text!="" && IsNumber(A23->Text) && A24->Text!="" &&
IsNumber(A24->Text) && A25->Text!="" && IsNumber(A25->Text) &&
A26->Text!="" && IsNumber(A26->Text) && A27->Text!="" &&
IsNumber(A27->Text) && A31->Text!="" && IsNumber(A31->Text) &&
A32->Text!="" && IsNumber(A32->Text) && A33->Text!="" &&
IsNumber(A33->Text) && A34->Text!="" && IsNumber(A34->Text) &&
A35->Text!="" && IsNumber(A35->Text) && A36->Text!="" &&
IsNumber(A36->Text) && A37->Text!="" && IsNumber(A37->Text) &&
A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
A44->Text!="" && IsNumber(A44->Text) && A45->Text!="" &&
IsNumber(A45->Text) && A46->Text!="" && IsNumber(A46->Text) &&
A47->Text!="" && IsNumber(A47->Text) && A51->Text!="" &&
IsNumber(A51->Text) && A52->Text!="" && IsNumber(A52->Text) &&
A53->Text!="" && IsNumber(A53->Text) && A54->Text!="" &&
IsNumber(A54->Text) && A55->Text!="" && IsNumber(A55->Text) &&
A56->Text!="" && IsNumber(A56->Text) && A57->Text!="" &&
IsNumber(A57->Text) && A61->Text!="" && IsNumber(A61->Text) &&
A62->Text!="" && IsNumber(A62->Text) && A63->Text!="" &&
IsNumber(A63->Text) && A64->Text!="" && IsNumber(A64->Text) &&
A65->Text!="" && IsNumber(A65->Text) && A66->Text!="" &&
IsNumber(A66->Text) && A67->Text!="" && IsNumber(A67->Text) &&
A71->Text!="" && IsNumber(A71->Text) && A72->Text!="" &&
IsNumber(A72->Text) && A73->Text!="" && IsNumber(A73->Text) &&
A74->Text!="" && IsNumber(A74->Text) && A75->Text!="" &&
IsNumber(A75->Text) && A76->Text!="" && IsNumber(A76->Text) &&
A77->Text!="" && IsNumber(A77->Text) && X1->Text!="" &&
IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
X3->Text!="" && IsNumber(X3->Text) && X4->Text!="" &&
IsNumber(X4->Text) && X5->Text!="" && IsNumber(X5->Text) &&
X6->Text!="" && IsNumber(X6->Text) && X7->Text!="" &&
IsNumber(X7->Text) && B1->Text!="" && IsNumber(B1->Text) &&
B2->Text!="" && IsNumber(B2->Text) && B3->Text!="" &&
IsNumber(B3->Text) && B4->Text!="" && IsNumber(B4->Text) &&
B5->Text!="" && IsNumber(B5->Text) && B6->Text!="" &&
IsNumber(B6->Text) && B7->Text!="" && IsNumber(B7->Text))
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[0][4]=A15->Text.ToDouble();
A[0][5]=A16->Text.ToDouble();
A[0][6]=A17->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[1][4]=A25->Text.ToDouble();
A[1][5]=A26->Text.ToDouble();
A[1][6]=A27->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();

```

```

A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
A[2][3]=A34->Text.ToDouble();
A[2][4]=A35->Text.ToDouble();
A[2][5]=A36->Text.ToDouble();
A[2][6]=A37->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();
A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
A[3][3]=A44->Text.ToDouble();
A[3][4]=A45->Text.ToDouble();
A[3][5]=A46->Text.ToDouble();
A[3][6]=A47->Text.ToDouble();
A[4][0]=A51->Text.ToDouble();
A[4][1]=A52->Text.ToDouble();
A[4][2]=A53->Text.ToDouble();
A[4][3]=A54->Text.ToDouble();
A[4][4]=A55->Text.ToDouble();
A[4][5]=A56->Text.ToDouble();
A[4][6]=A57->Text.ToDouble();
A[5][0]=A61->Text.ToDouble();
A[5][1]=A62->Text.ToDouble();
A[5][2]=A63->Text.ToDouble();
A[5][3]=A64->Text.ToDouble();
A[5][4]=A65->Text.ToDouble();
A[5][5]=A66->Text.ToDouble();
A[5][6]=A67->Text.ToDouble();
A[6][0]=A71->Text.ToDouble();
A[6][1]=A72->Text.ToDouble();
A[6][2]=A73->Text.ToDouble();
A[6][3]=A74->Text.ToDouble();
A[6][4]=A75->Text.ToDouble();
A[6][5]=A76->Text.ToDouble();
A[6][6]=A77->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
X[4]=X5->Text.ToDouble();
X[5]=X6->Text.ToDouble();
X[6]=X7->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();
B[4]=B5->Text.ToDouble();
B[5]=B6->Text.ToDouble();
B[6]=B7->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
            MB_OK|MB_ICONSTOP);
}
if (EditsFull)
{
do
{norma=0;
for (i=0;i<N;i++)
FX[i]=0;
for (i=0;i<N;i++)
{
for (j=0;j<N;j++)
{
if (i!=j && A[i][i]!=0)
FX[i]=FX[i]-X[j]*(A[i][j]/A[i][i]);
}
if (A[i][i]!=0)
FX[i]=FX[i]+(B[i]/A[i][i]);
else
{
MessageBox(NULL,"Υπολογισμός Αδύνατος! Έξοδος!", "ΣΦΑΛΜΑ", MB_OK|MB_ICONSTOP);
break;
}
}
norma=norma+(FX[i]-X[i])*(FX[i]-X[i]);
}
for (i=0;i<N;i++)

```

```

    X[i]=FX[i];
}while (sqrt (norma)<limit);
FX1->Text=FX[0];
FX2->Text=FX[1];
FX3->Text=FX[2];
FX4->Text=FX[3];
FX5->Text=FX[4];
FX6->Text=FX[5];
FX7->Text=FX[6];
}
}

if (N==1)
{
    FX1->Enabled=true;
    FX2->Text="";    FX2->Enabled=false;
    FX3->Text="";    FX3->Enabled=false;
    FX4->Text="";    FX4->Enabled=false;
    FX5->Text="";    FX5->Enabled=false;
    FX6->Text="";    FX6->Enabled=false;
    FX7->Text="";    FX7->Enabled=false;
}

else if (N==2)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Text="";    FX3->Enabled=false;
    FX4->Text="";    FX4->Enabled=false;
    FX5->Text="";    FX5->Enabled=false;
    FX6->Text="";    FX6->Enabled=false;
    FX7->Text="";    FX7->Enabled=false;
}

else if (N==3)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Text="";    FX4->Enabled=false;
    FX5->Text="";    FX5->Enabled=false;
    FX6->Text="";    FX6->Enabled=false;
    FX7->Text="";    FX7->Enabled=false;
}

else if (N==4)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Text="";    FX5->Enabled=false;
    FX6->Text="";    FX6->Enabled=false;
    FX7->Text="";    FX7->Enabled=false;
}

else if (N==5)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Enabled=true;
    FX6->Text="";    FX6->Enabled=false;
    FX7->Text="";    FX7->Enabled=false;
}

else if (N==6)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Enabled=true;
    FX6->Enabled=true;
    FX7->Text="";    FX7->Enabled=false;
}

```

```

else if (N==7)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Enabled=true;
    FX6->Enabled=true;
    FX7->Enabled=true;
}

else
MessageBox(NULL,"Μη αποδεκτός Χαρακτήρας!", "Σφάλμα", MB_OK| MB_ICONSTOP);
}
//-----
void __fastcall TForm12::N6Click(TObject *Sender)
{
    MethHelpForm->Show();
    MethHelpForm->TabSheet2->Show();
    MethHelpForm->BitBtn6->Click();
}
//-----
void __fastcall TForm12::N7Click(TObject *Sender)
{
    ProgHelpForm->Show();
    ProgHelpForm->BitBtn3->Click();
    ProgHelpForm->Button2->Click();
}

```

## Unit 13 (Μέθοδος Gauss - Seidel)

```
#include<math.h>
#include <vcl.h>
#pragma hdrstop
#include "Unit13.h"
#include "Unit9.h"
#include "Unit5.h"
#include "Unit11.h"
#include "Unit14.h"
#include "Unit15.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm13 *Form13;
//-----
__fastcall TForm13::TForm13(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i])&& edt[i]!='-'&& edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm13::N8Click(TObject *Sender)
{
  AboutForm->ShowModal();
}
//-----
void __fastcall TForm13::N4Click(TObject *Sender)
{
  Form13->Close();
}
//-----
void __fastcall TForm13::FormClose(TObject*Sender,TCloseAction &Action)
{
  Application->Terminate();
}
//-----
void __fastcall TForm13::N3Click(TObject *Sender)
{
  Form13->Hide();
  Form11->Show();
}
//-----
void __fastcall TForm13::Edit1Change(TObject *Sender)
{
  Edit2->Text=Edit1->Text;
  Edit4->Text=Edit1->Text;
  BitBtn1->Caption="OK";
  //Reset
  A11->Enabled=false; A11->Color=clSkyBlue;
  A12->Enabled=false; A12->Color=clSkyBlue;
  A13->Enabled=false; A13->Color=clSkyBlue;
  A14->Enabled=false; A14->Color=clSkyBlue;
  A15->Enabled=false; A15->Color=clSkyBlue;
  A16->Enabled=false; A16->Color=clSkyBlue;
  A17->Enabled=false; A17->Color=clSkyBlue;
  A21->Enabled=false; A21->Color=clSkyBlue;
  A22->Enabled=false; A22->Color=clSkyBlue;
  A23->Enabled=false; A23->Color=clSkyBlue;
  A24->Enabled=false; A24->Color=clSkyBlue;
  A25->Enabled=false; A25->Color=clSkyBlue;
  A26->Enabled=false; A26->Color=clSkyBlue;
  A27->Enabled=false; A27->Color=clSkyBlue;
  A31->Enabled=false; A31->Color=clSkyBlue;
  A32->Enabled=false; A32->Color=clSkyBlue;
```

A33->Enabled=**false**; A33->Color=clSkyBlue;  
A34->Enabled=**false**; A34->Color=clSkyBlue;  
A35->Enabled=**false**; A35->Color=clSkyBlue;  
A36->Enabled=**false**; A36->Color=clSkyBlue;  
A37->Enabled=**false**; A37->Color=clSkyBlue;  
A41->Enabled=**false**; A41->Color=clSkyBlue;  
A42->Enabled=**false**; A42->Color=clSkyBlue;  
A43->Enabled=**false**; A43->Color=clSkyBlue;  
A44->Enabled=**false**; A44->Color=clSkyBlue;  
A45->Enabled=**false**; A45->Color=clSkyBlue;  
A46->Enabled=**false**; A46->Color=clSkyBlue;  
A47->Enabled=**false**; A47->Color=clSkyBlue;  
A51->Enabled=**false**; A51->Color=clSkyBlue;  
A52->Enabled=**false**; A52->Color=clSkyBlue;  
A53->Enabled=**false**; A53->Color=clSkyBlue;  
A54->Enabled=**false**; A54->Color=clSkyBlue;  
A55->Enabled=**false**; A55->Color=clSkyBlue;  
A56->Enabled=**false**; A56->Color=clSkyBlue;  
A57->Enabled=**false**; A57->Color=clSkyBlue;  
A61->Enabled=**false**; A61->Color=clSkyBlue;  
A62->Enabled=**false**; A62->Color=clSkyBlue;  
A63->Enabled=**false**; A63->Color=clSkyBlue;  
A64->Enabled=**false**; A64->Color=clSkyBlue;  
A65->Enabled=**false**; A65->Color=clSkyBlue;  
A66->Enabled=**false**; A66->Color=clSkyBlue;  
A67->Enabled=**false**; A67->Color=clSkyBlue;  
A71->Enabled=**false**; A71->Color=clSkyBlue;  
A72->Enabled=**false**; A72->Color=clSkyBlue;  
A73->Enabled=**false**; A73->Color=clSkyBlue;  
A74->Enabled=**false**; A74->Color=clSkyBlue;  
A75->Enabled=**false**; A75->Color=clSkyBlue;  
A76->Enabled=**false**; A76->Color=clSkyBlue;  
A77->Enabled=**false**; A77->Color=clSkyBlue;  
X1->Enabled=**false**; X1->Color=clSkyBlue;  
X2->Enabled=**false**; X2->Color=clSkyBlue;  
X3->Enabled=**false**; X3->Color=clSkyBlue;  
X4->Enabled=**false**; X4->Color=clSkyBlue;  
X5->Enabled=**false**; X5->Color=clSkyBlue;  
X6->Enabled=**false**; X6->Color=clSkyBlue;  
X7->Enabled=**false**; X7->Color=clSkyBlue;  
B1->Enabled=**false**; B1->Color=clSkyBlue;  
B2->Enabled=**false**; B2->Color=clSkyBlue;  
B3->Enabled=**false**; B3->Color=clSkyBlue;  
B4->Enabled=**false**; B4->Color=clSkyBlue;  
B5->Enabled=**false**; B5->Color=clSkyBlue;  
B6->Enabled=**false**; B6->Color=clSkyBlue;  
B7->Enabled=**false**; B7->Color=clSkyBlue;  
  
A11->Text="";  
A12->Text="";  
A13->Text="";  
A14->Text="";  
A15->Text="";  
A16->Text="";  
A17->Text="";  
A21->Text="";  
A22->Text="";  
A23->Text="";  
A24->Text="";  
A25->Text="";  
A26->Text="";  
A27->Text="";  
A31->Text="";  
A32->Text="";  
A33->Text="";  
A34->Text="";  
A35->Text="";  
A36->Text="";  
A37->Text="";  
A41->Text="";  
A42->Text="";  
A43->Text="";  
A44->Text="";  
A45->Text="";  
A46->Text="";  
A47->Text="";  
A51->Text="";

```

A52->Text="";
A53->Text="";
A54->Text="";
A55->Text="";
A56->Text="";
A57->Text="";
A61->Text="";
A62->Text="";
A63->Text="";
A64->Text="";
A65->Text="";
A66->Text="";
A67->Text="";
A71->Text="";
A72->Text="";
A73->Text="";
A74->Text="";
A75->Text="";
A76->Text="";
A77->Text="";
X1->Text="";
X2->Text="";
X3->Text="";
X4->Text="";
X5->Text="";
X6->Text="";
X7->Text="";
B1->Text="";
B2->Text="";
B3->Text="";
B4->Text="";
B5->Text="";
B6->Text="";
B7->Text="";
}
//-----
void __fastcall TForm13::BitBtn1Click(TObject *Sender)
{
    if(IsNumber(Edit1->Text) && IsNumber(Edit2->Text))
    {
        if(Edit1->Text==1)
        {
            A11->Enabled=true; A11->Color=clCream;
            X1->Enabled=true; X1->Color=clCream;
            B1->Enabled=true; B1->Color=clCream;
            BitBtn1->Caption="Ready";
            Label10->Enabled=true;
        }
        else
        if(Edit1->Text==2)
        {
            A11->Enabled=true; A11->Color=clCream;
            A12->Enabled=true; A12->Color=clCream;
            A21->Enabled=true; A21->Color=clCream;
            A22->Enabled=true; A22->Color=clCream;
            X1->Enabled=true; X1->Color=clCream;
            X2->Enabled=true; X2->Color=clCream;
            B1->Enabled=true; B1->Color=clCream;
            B2->Enabled=true; B2->Color=clCream;
            BitBtn1->Caption="Ready";
            Label10->Enabled=true;
        }
        else
        if(Edit1->Text==3)
        {
            A11->Enabled=true; A11->Color=clCream;
            A12->Enabled=true; A12->Color=clCream;
            A13->Enabled=true; A13->Color=clCream;
            A21->Enabled=true; A21->Color=clCream;
            A22->Enabled=true; A22->Color=clCream;
            A23->Enabled=true; A23->Color=clCream;
            A31->Enabled=true; A31->Color=clCream;
            A32->Enabled=true; A32->Color=clCream;
            A33->Enabled=true; A33->Color=clCream;
            X1->Enabled=true; X1->Color=clCream;
            X2->Enabled=true; X2->Color=clCream;
            X3->Enabled=true; X3->Color=clCream;
        }
    }
}

```



```

B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
if(Edit1->Text==4)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;
A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
if(Edit1->Text==5)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;
A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A15->Enabled=true; A15->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A25->Enabled=true; A25->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A35->Enabled=true; A35->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
A45->Enabled=true; A45->Color=clCream;
A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
}

```

```

else
if (Edit1->Text==6)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;
A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A15->Enabled=true; A15->Color=clCream;
A16->Enabled=true; A16->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A25->Enabled=true; A25->Color=clCream;
A26->Enabled=true; A26->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A35->Enabled=true; A35->Color=clCream;
A36->Enabled=true; A36->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
A45->Enabled=true; A45->Color=clCream;
A46->Enabled=true; A46->Color=clCream;
A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
A56->Enabled=true; A56->Color=clCream;
A61->Enabled=true; A61->Color=clCream;
A62->Enabled=true; A62->Color=clCream;
A63->Enabled=true; A63->Color=clCream;
A64->Enabled=true; A64->Color=clCream;
A65->Enabled=true; A65->Color=clCream;
A66->Enabled=true; A66->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
X6->Enabled=true; X6->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
B6->Enabled=true; B6->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
if (Edit1->Text==7)
{
A11->Enabled=true; A11->Color=clCream;
A12->Enabled=true; A12->Color=clCream;
A13->Enabled=true; A13->Color=clCream;
A14->Enabled=true; A14->Color=clCream;
A15->Enabled=true; A15->Color=clCream;
A16->Enabled=true; A16->Color=clCream;
A17->Enabled=true; A17->Color=clCream;
A21->Enabled=true; A21->Color=clCream;
A22->Enabled=true; A22->Color=clCream;
A23->Enabled=true; A23->Color=clCream;
A24->Enabled=true; A24->Color=clCream;
A25->Enabled=true; A25->Color=clCream;
A26->Enabled=true; A26->Color=clCream;
A27->Enabled=true; A27->Color=clCream;
A31->Enabled=true; A31->Color=clCream;
A32->Enabled=true; A32->Color=clCream;
A33->Enabled=true; A33->Color=clCream;
A34->Enabled=true; A34->Color=clCream;
A35->Enabled=true; A35->Color=clCream;
A36->Enabled=true; A36->Color=clCream;

```

```

A37->Enabled=true; A37->Color=clCream;
A41->Enabled=true; A41->Color=clCream;
A42->Enabled=true; A42->Color=clCream;
A43->Enabled=true; A43->Color=clCream;
A44->Enabled=true; A44->Color=clCream;
A45->Enabled=true; A45->Color=clCream;
A46->Enabled=true; A46->Color=clCream;
A47->Enabled=true; A47->Color=clCream;
A51->Enabled=true; A51->Color=clCream;
A52->Enabled=true; A52->Color=clCream;
A53->Enabled=true; A53->Color=clCream;
A54->Enabled=true; A54->Color=clCream;
A55->Enabled=true; A55->Color=clCream;
A56->Enabled=true; A56->Color=clCream;
A57->Enabled=true; A57->Color=clCream;
A61->Enabled=true; A61->Color=clCream;
A62->Enabled=true; A62->Color=clCream;
A63->Enabled=true; A63->Color=clCream;
A64->Enabled=true; A64->Color=clCream;
A65->Enabled=true; A65->Color=clCream;
A66->Enabled=true; A66->Color=clCream;
A67->Enabled=true; A67->Color=clCream;
A71->Enabled=true; A71->Color=clCream;
A72->Enabled=true; A72->Color=clCream;
A73->Enabled=true; A73->Color=clCream;
A74->Enabled=true; A74->Color=clCream;
A75->Enabled=true; A75->Color=clCream;
A76->Enabled=true; A76->Color=clCream;
A77->Enabled=true; A77->Color=clCream;
X1->Enabled=true; X1->Color=clCream;
X2->Enabled=true; X2->Color=clCream;
X3->Enabled=true; X3->Color=clCream;
X4->Enabled=true; X4->Color=clCream;
X5->Enabled=true; X5->Color=clCream;
X6->Enabled=true; X6->Color=clCream;
X7->Enabled=true; X7->Color=clCream;
B1->Enabled=true; B1->Color=clCream;
B2->Enabled=true; B2->Color=clCream;
B3->Enabled=true; B3->Color=clCream;
B4->Enabled=true; B4->Color=clCream;
B5->Enabled=true; B5->Color=clCream;
B6->Enabled=true; B6->Color=clCream;
B7->Enabled=true; B7->Color=clCream;
BitBtn1->Caption="Ready";
Label10->Enabled=true;
}
else
    MessageBox(NULL, "Λάθος Δεδομένα!", "Σφάλμα", MB_OK|MB_ICONSTOP);
}
else
    MessageBox(NULL, "Μη αποδεκτός Χαρακτήρας!", "Σφάλμα", MB_OK| MB_ICONSTOP);
}
//-----
void __fastcall TForm13::N2Click(TObject *Sender)
{
    BOOL EditsFull=false;
    int i, j, N=0;
    double A[7][7], X[7], B[7], FX[7], norma, limit;
    for (i=0; i<7; i++)
        for (j=0; j<7; j++)
            {
                A[i][j]=0;
                X[i]=0;
                B[i]=0;
                FX[i]=0;
            }
    if (Label10->Enabled && IsNumber(Edit3->Text) && Edit3->Text!="")
        {
            limit=Edit3->Text.ToDouble();
            N=Edit4->Text.ToInt();
            if (N==2)
                {
                    if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
                        IsNumber(A12->Text) && A21->Text!="" && IsNumber(A21->Text) &&
                        A22->Text!="" && IsNumber(A22->Text) && X1->Text!="" &&
                        IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
                        B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
                        IsNumber(B2->Text))
                        {
                            A[0][0]=A11->Text.ToDouble();
                            A[0][1]=A12->Text.ToDouble();

```

```

A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if (N==3)
{if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A21->Text!="" && IsNumber(A21->Text) && A22->Text!="" &&
IsNumber(A22->Text) && A23->Text!="" && IsNumber(A23->Text) &&
A31->Text!="" && IsNumber(A31->Text) && A32->Text!="" &&
IsNumber(A32->Text) && A33->Text!="" && IsNumber(A33->Text) &&
X1->Text!="" && IsNumber(X1->Text) && X2->Text!="" &&
IsNumber(X2->Text) && X3->Text!="" && IsNumber(X3->Text) &&
B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
IsNumber(B2->Text) && B3->Text!="" && IsNumber(B3->Text))
{A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if (N==4)
{if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A14->Text!="" && IsNumber(A14->Text) && A21->Text!="" &&
IsNumber(A21->Text) && A22->Text!="" && IsNumber(A22->Text) &&
A23->Text!="" && IsNumber(A23->Text) && A24->Text!="" &&
IsNumber(A24->Text) && A31->Text!="" && IsNumber(A31->Text) &&
A32->Text!="" && IsNumber(A32->Text) && A33->Text!="" &&
IsNumber(A33->Text) && A34->Text!="" && IsNumber(A34->Text) &&
A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
A44->Text!="" && IsNumber(A44->Text) && X1->Text!="" &&
IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
X3->Text!="" && IsNumber(X3->Text) && X4->Text!="" &&
IsNumber(X4->Text) && B1->Text!="" && IsNumber(B1->Text) &&
B2->Text!="" && IsNumber(B2->Text) && B3->Text!="" &&
IsNumber(B3->Text) && B4->Text!="" && IsNumber(B4->Text))
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
A[2][3]=A34->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();
A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
}
}

```



```

}
else
MessageBox(NULL, "Ελεγιπή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
    MB_OK|MB_ICONSTOP);
}
if (N==6)
{if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
    IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
    A14->Text!="" && IsNumber(A14->Text) && A15->Text!="" &&
    IsNumber(A15->Text) && A16->Text!="" && IsNumber(A16->Text) &&
    A21->Text!="" && IsNumber(A21->Text) && A22->Text!="" &&
    IsNumber(A22->Text) && A23->Text!="" && IsNumber(A23->Text) &&
    A24->Text!="" && IsNumber(A24->Text) && A25->Text!="" &&
    IsNumber(A25->Text) && A26->Text!="" && IsNumber(A26->Text) &&
    A31->Text!="" && IsNumber(A31->Text) && A32->Text!="" &&
    IsNumber(A32->Text) && A33->Text!="" && IsNumber(A33->Text) &&
    A34->Text!="" && IsNumber(A34->Text) && A35->Text!="" &&
    IsNumber(A35->Text) && A36->Text!="" && IsNumber(A36->Text) &&
    A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
    IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
    A44->Text!="" && IsNumber(A44->Text) && A45->Text!="" &&
    IsNumber(A45->Text) && A46->Text!="" && IsNumber(A46->Text) &&
    A51->Text!="" && IsNumber(A51->Text) && A52->Text!="" &&
    IsNumber(A52->Text) && A53->Text!="" && IsNumber(A53->Text) &&
    A54->Text!="" && IsNumber(A54->Text) && A55->Text!="" &&
    IsNumber(A55->Text) && A56->Text!="" && IsNumber(A56->Text) &&
    A61->Text!="" && IsNumber(A61->Text) && A62->Text!="" &&
    IsNumber(A62->Text) && A63->Text!="" && IsNumber(A63->Text) &&
    A64->Text!="" && IsNumber(A64->Text) && A65->Text!="" &&
    IsNumber(A65->Text) && A66->Text!="" && IsNumber(A66->Text) &&
    X1->Text!="" && IsNumber(X1->Text) && X2->Text!="" &&
    IsNumber(X2->Text) && X3->Text!="" && IsNumber(X3->Text) &&
    X4->Text!="" && IsNumber(X4->Text) && X5->Text!="" &&
    IsNumber(X5->Text) && X6->Text!="" && IsNumber(X6->Text) &&
    B1->Text!="" && IsNumber(B1->Text) && B2->Text!="" &&
    IsNumber(B2->Text) && B3->Text!="" && IsNumber(B3->Text) &&
    B4->Text!="" && IsNumber(B4->Text) && B5->Text!="" &&
    IsNumber(B5->Text) && B6->Text!="" && IsNumber(B6->Text)
)
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[0][4]=A15->Text.ToDouble();
A[0][5]=A16->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[1][4]=A25->Text.ToDouble();
A[1][5]=A26->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();
A[2][3]=A34->Text.ToDouble();
A[2][4]=A35->Text.ToDouble();
A[2][5]=A36->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();
A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
A[3][3]=A44->Text.ToDouble();
A[3][4]=A45->Text.ToDouble();
A[3][5]=A46->Text.ToDouble();
A[4][0]=A51->Text.ToDouble();
A[4][1]=A52->Text.ToDouble();
A[4][2]=A53->Text.ToDouble();
A[4][3]=A54->Text.ToDouble();
A[4][4]=A55->Text.ToDouble();
A[4][5]=A56->Text.ToDouble();
A[5][0]=A61->Text.ToDouble();
A[5][1]=A62->Text.ToDouble();
A[5][2]=A63->Text.ToDouble();
A[5][3]=A64->Text.ToDouble();
A[5][4]=A65->Text.ToDouble();
A[5][5]=A66->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();

```

```

X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
X[4]=X5->Text.ToDouble();
X[5]=X6->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();
B[4]=B5->Text.ToDouble();
B[5]=B6->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεειπή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
MB_OK|MB_ICONSTOP);
}
if (N==7)
{ if (A11->Text!="" && IsNumber(A11->Text) && A12->Text!="" &&
IsNumber(A12->Text) && A13->Text!="" && IsNumber(A13->Text) &&
A14->Text!="" && IsNumber(A14->Text) && A15->Text!="" &&
IsNumber(A15->Text) && A16->Text!="" && IsNumber(A16->Text) &&
A17->Text!="" && IsNumber(A17->Text) && A21->Text!="" &&
IsNumber(A21->Text) && A22->Text!="" && IsNumber(A22->Text) &&
A23->Text!="" && IsNumber(A23->Text) && A24->Text!="" &&
IsNumber(A24->Text) && A25->Text!="" && IsNumber(A25->Text) &&
A26->Text!="" && IsNumber(A26->Text) && A27->Text!="" &&
IsNumber(A27->Text) && A31->Text!="" && IsNumber(A31->Text) &&
A32->Text!="" && IsNumber(A32->Text) && A33->Text!="" &&
IsNumber(A33->Text) && A34->Text!="" && IsNumber(A34->Text) &&
A35->Text!="" && IsNumber(A35->Text) && A36->Text!="" &&
IsNumber(A36->Text) && A37->Text!="" && IsNumber(A37->Text) &&
A41->Text!="" && IsNumber(A41->Text) && A42->Text!="" &&
IsNumber(A42->Text) && A43->Text!="" && IsNumber(A43->Text) &&
A44->Text!="" && IsNumber(A44->Text) && A45->Text!="" &&
IsNumber(A45->Text) && A46->Text!="" && IsNumber(A46->Text) &&
A47->Text!="" && IsNumber(A47->Text) && A51->Text!="" &&
IsNumber(A51->Text) && A52->Text!="" && IsNumber(A52->Text) &&
A53->Text!="" && IsNumber(A53->Text) && A54->Text!="" &&
IsNumber(A54->Text) && A55->Text!="" && IsNumber(A55->Text) &&
A56->Text!="" && IsNumber(A56->Text) && A57->Text!="" &&
IsNumber(A57->Text) && A61->Text!="" && IsNumber(A61->Text) &&
A62->Text!="" && IsNumber(A62->Text) && A63->Text!="" &&
IsNumber(A63->Text) && A64->Text!="" && IsNumber(A64->Text) &&
A65->Text!="" && IsNumber(A65->Text) && A66->Text!="" &&
IsNumber(A66->Text) && A67->Text!="" && IsNumber(A67->Text) &&
A71->Text!="" && IsNumber(A71->Text) && A72->Text!="" &&
IsNumber(A72->Text) && A73->Text!="" && IsNumber(A73->Text) &&
A74->Text!="" && IsNumber(A74->Text) && A75->Text!="" &&
IsNumber(A75->Text) && A76->Text!="" && IsNumber(A76->Text) &&
A77->Text!="" && IsNumber(A77->Text) && X1->Text!="" &&
IsNumber(X1->Text) && X2->Text!="" && IsNumber(X2->Text) &&
X3->Text!="" && IsNumber(X3->Text) && X4->Text!="" &&
IsNumber(X4->Text) && X5->Text!="" && IsNumber(X5->Text) &&
X6->Text!="" && IsNumber(X6->Text) && X7->Text!="" &&
IsNumber(X7->Text) && B1->Text!="" && IsNumber(B1->Text) &&
B2->Text!="" && IsNumber(B2->Text) && B3->Text!="" &&
IsNumber(B3->Text) && B4->Text!="" && IsNumber(B4->Text) &&
B5->Text!="" && IsNumber(B5->Text) && B6->Text!="" &&
IsNumber(B6->Text) && B7->Text!="" && IsNumber(B7->Text))
{
A[0][0]=A11->Text.ToDouble();
A[0][1]=A12->Text.ToDouble();
A[0][2]=A13->Text.ToDouble();
A[0][3]=A14->Text.ToDouble();
A[0][4]=A15->Text.ToDouble();
A[0][5]=A16->Text.ToDouble();
A[0][6]=A17->Text.ToDouble();
A[1][0]=A21->Text.ToDouble();
A[1][1]=A22->Text.ToDouble();
A[1][2]=A23->Text.ToDouble();
A[1][3]=A24->Text.ToDouble();
A[1][4]=A25->Text.ToDouble();
A[1][5]=A26->Text.ToDouble();
A[1][6]=A27->Text.ToDouble();
A[2][0]=A31->Text.ToDouble();
A[2][1]=A32->Text.ToDouble();
A[2][2]=A33->Text.ToDouble();

```

```

A[2][3]=A34->Text.ToDouble();
A[2][4]=A35->Text.ToDouble();
A[2][5]=A36->Text.ToDouble();
A[2][6]=A37->Text.ToDouble();
A[3][0]=A41->Text.ToDouble();
A[3][1]=A42->Text.ToDouble();
A[3][2]=A43->Text.ToDouble();
A[3][3]=A44->Text.ToDouble();
A[3][4]=A45->Text.ToDouble();
A[3][5]=A46->Text.ToDouble();
A[3][6]=A47->Text.ToDouble();
A[4][0]=A51->Text.ToDouble();
A[4][1]=A52->Text.ToDouble();
A[4][2]=A53->Text.ToDouble();
A[4][3]=A54->Text.ToDouble();
A[4][4]=A55->Text.ToDouble();
A[4][5]=A56->Text.ToDouble();
A[4][6]=A57->Text.ToDouble();
A[5][0]=A61->Text.ToDouble();
A[5][1]=A62->Text.ToDouble();
A[5][2]=A63->Text.ToDouble();
A[5][3]=A64->Text.ToDouble();
A[5][4]=A65->Text.ToDouble();
A[5][5]=A66->Text.ToDouble();
A[5][6]=A67->Text.ToDouble();
A[6][0]=A71->Text.ToDouble();
A[6][1]=A72->Text.ToDouble();
A[6][2]=A73->Text.ToDouble();
A[6][3]=A74->Text.ToDouble();
A[6][4]=A75->Text.ToDouble();
A[6][5]=A76->Text.ToDouble();
A[6][6]=A77->Text.ToDouble();
X[0]=X1->Text.ToDouble();
X[1]=X2->Text.ToDouble();
X[2]=X3->Text.ToDouble();
X[3]=X4->Text.ToDouble();
X[4]=X5->Text.ToDouble();
X[5]=X6->Text.ToDouble();
X[6]=X7->Text.ToDouble();
B[0]=B1->Text.ToDouble();
B[1]=B2->Text.ToDouble();
B[2]=B3->Text.ToDouble();
B[3]=B4->Text.ToDouble();
B[4]=B5->Text.ToDouble();
B[5]=B6->Text.ToDouble();
B[6]=B7->Text.ToDouble();
EditsFull=true;
}
else
MessageBox(NULL,"Ελεεινή ή Λάθος Στοιχεία Στους Πίνακες!", "Σφάλμα",
            MB_OK|MB_ICONSTOP);
}
if(EditsFull)
{
do
{norma=0;
for(i=0;i<N;i++)
FX[i]=0;
for(i=0;i<N;i++)
{
for(j=0;j<N;j++)
{
if(i!=j && A[i][i]!=0)
FX[i]=FX[i]-X[j]*(A[i][j]/A[i][i]);
}
if(A[i][i]!=0)
{
FX[i]=FX[i]+(B[i]/A[i][i]);
norma=norma+(FX[i]-X[i])*(FX[i]-X[i]);
X[i]=FX[i];
}
else
{
MessageBox(NULL,"Υπολογισμός Αδύνατος!Εξοδος!", "ΣΦΑΛΜΑ", MB_OK | MB_ICONSTOP);

break;
}
}
}

```



```

    }
}while(sqrt(norma)<limit);
FX1->Text=FX[0];
FX2->Text=FX[1];
FX3->Text=FX[2];
FX4->Text=FX[3];
FX5->Text=FX[4];
FX6->Text=FX[5];
FX7->Text=FX[6];
}
}

if (N==1)
{
    FX1->Enabled=true;
    FX2->Text="";   FX2->Enabled=false;
    FX3->Text="";   FX3->Enabled=false;
    FX4->Text="";   FX4->Enabled=false;
    FX5->Text="";   FX5->Enabled=false;
    FX6->Text="";   FX6->Enabled=false;
    FX7->Text="";   FX7->Enabled=false;
}

else if (N==2)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Text="";   FX3->Enabled=false;
    FX4->Text="";   FX4->Enabled=false;
    FX5->Text="";   FX5->Enabled=false;
    FX6->Text="";   FX6->Enabled=false;
    FX7->Text="";   FX7->Enabled=false;
}

else if (N==3)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Text="";   FX4->Enabled=false;
    FX5->Text="";   FX5->Enabled=false;
    FX6->Text="";   FX6->Enabled=false;
    FX7->Text="";   FX7->Enabled=false;
}

else if (N==4)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Text="";   FX5->Enabled=false;
    FX6->Text="";   FX6->Enabled=false;
    FX7->Text="";   FX7->Enabled=false;
}

else if (N==5)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Enabled=true;
    FX6->Text="";   FX6->Enabled=false;
    FX7->Text="";   FX7->Enabled=false;
}

else if (N==6)
{
    FX1->Enabled=true;
    FX2->Enabled=true;
    FX3->Enabled=true;
    FX4->Enabled=true;
    FX5->Enabled=true;
    FX6->Enabled=true;
    FX7->Text="";   FX7->Enabled=false;
}

```

```

else if (N==7)
{
FX1->Enabled=true;
FX2->Enabled=true;
FX3->Enabled=true;
FX4->Enabled=true;
FX5->Enabled=true;
FX6->Enabled=true;
FX7->Enabled=true;
}

else
MessageBox(NULL,"Μη αποδεκτός Χαρακτήρας!", "Σφάλμα", MB_OK| MB_ICONSTOP);
}
//-----
void __fastcall TForm13::N6Click(TObject *Sender)
{
MethHelpForm->Show();
MethHelpForm->TabSheet2->Show();
MethHelpForm->BitBtn7->Click();
}
//-----
void __fastcall TForm13::N7Click(TObject *Sender)
{
ProgHelpForm->Show();
ProgHelpForm->BitBtn3->Click();
ProgHelpForm->Button2->Click();
}

```

## Unit 14 (Βοήθεια χρήσης)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit14.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TProgHelpForm *ProgHelpForm;
//-----
__fastcall TProgHelpForm::TProgHelpForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TProgHelpForm::BitBtn6Click(TObject *Sender)
{
    Mem1->Clear();
    ProgHelpForm->Close();
}
//-----
void __fastcall TProgHelpForm::BitBtn1Click(TObject *Sender)
{
    Mem1->Clear();
    Button1->Enabled=false;
    Button2->Enabled=false;
    Button3->Enabled=false;
    Mem1->Lines->LoadFromFile("Data//Introduction.txt");
}
//-----
void __fastcall TProgHelpForm::BitBtn2Click(TObject *Sender)
{
    Mem1->Clear();
    Button1->Enabled=false;
    Button2->Enabled=false;
    Button3->Enabled=false;
    Mem1->Lines->LoadFromFile("Data//Askhseis.txt");
}
//-----
void __fastcall TProgHelpForm::BitBtn3Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data//Methodoi.txt");
    Button1->Enabled=true;
    Button2->Enabled=true;
    Button3->Enabled=true;
}
//-----
void __fastcall TProgHelpForm::Button1Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data//MhGrammikes.txt");
}
//-----
void __fastcall TProgHelpForm::Button2Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data\\Grammikes.txt");
}
//-----
void __fastcall TProgHelpForm::BitBtn7Click(TObject *Sender)
{
    Mem1->Clear();
    Button1->Enabled=false;
    Button2->Enabled=false;
    Button3->Enabled=false;
    Mem1->Lines->LoadFromFile("Data//Synarthsh.txt");
}
//-----
void __fastcall TProgHelpForm::BitBtn4Click(TObject *Sender)
{
    Mem1->Clear();
    Button1->Enabled=false;
    Button2->Enabled=false;
    Button3->Enabled=false;
    Mem1->Lines->LoadFromFile("Data//Grafikh.txt");
}
```

```
}  
//-----  
void __fastcall TProgHelpForm::Button3Click(TObject *Sender)  
{  
    Mem1->Clear();  
    Mem1->Lines->LoadFromFile("Data\\Odtetragonon.txt");  
}
```

## Unit 15 (Βοήθεια μεθόδων)

```
#include <vcl.h>
#pragma hdrstop
#include "Unit15.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TMethHelpForm *MethHelpForm;
//-----
__fastcall TMethHelpForm::TMethHelpForm(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TMethHelpForm::BitBtn5Click(TObject *Sender)
{
    MethHelpForm->Close();
}
//-----
void __fastcall TMethHelpForm::BitBtn1Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data\\bisection.txt");
}
//-----
void __fastcall TMethHelpForm::BitBtn2Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data\\Regula Falsi_Temnousa.txt");
}
//-----
void __fastcall TMethHelpForm::BitBtn3Click(TObject *Sender)
{
    Mem1->Clear();
    Mem1->Lines->LoadFromFile("Data\\Newton.txt");
}
//-----
void __fastcall TMethHelpForm::BitBtn6Click(TObject *Sender)
{
    Memo2->Clear();
    Memo2->Lines->LoadFromFile("Data\\Jacobi.txt");
}
//-----
void __fastcall TMethHelpForm::BitBtn7Click(TObject *Sender)
{
    Memo2->Clear();
    Memo2->Lines->LoadFromFile("Data\\Gauss_Seidel.txt");
}
//-----
void __fastcall TMethHelpForm::TabSheet3Show(TObject *Sender)
{
    Memo3->Lines->LoadFromFile("Data\\tetragonon.txt");
}
}
```

## Unit 16 (Μέθοδος ελαχίστων τετραγώνων)

```
#include <math.h>
#include<math.hpp>
#include <vcl.h>
#pragma hdrstop
#include "Unit16.h"
#include "Unit2.h"
#include "Unit5.h"
#include "Unit14.h"
#include "Unit15.h"
#include "Unit9.h"
//-----
#pragma package (smart_init)
#pragma resource "*.dfm"
TForm16 *Form16;
TColor LineColor=clYellow;
TColor ImageCanBrush=clNavy;
TColor ImageCanPen=clMaroon;
//-----
__fastcall TForm16::TForm16(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
double func(double x)
{
    double y,a,b;
    a=Form16->Edit2->Text.ToDouble();
    b=Form16->Edit3->Text.ToDouble();
    y=a+b*x;
    return y;
}
//-----
BOOL IsNumber(AnsiString edt)
{int i;
  BOOL itis=true;
  for(i=1;i<=edt.Length();i++)
  { if(!isdigit(edt[i]) && edt[i]!='.' && edt[i]!='-' && edt[i]!='.')
    itis=false;
  }
  return itis;
}
//-----
void __fastcall TForm16::N10Click(TObject *Sender)
{
  Form16->Hide();
  ASKHSEIS->Show();
}
//-----
void __fastcall TForm16::N11Click(TObject *Sender)
{
  Application->Terminate();
}
//-----
void __fastcall TForm16::FormClose(TObject *Sender,
                                   TCloseAction &Action)
{
  Application->Terminate();
}
//-----
void __fastcall TForm16::N8Click(TObject *Sender)
{
  AboutForm->Show();
}
//-----
void __fastcall TForm16::N7Click(TObject *Sender)
{
  ProgHelpForm->Show();
  ProgHelpForm->BitBtn3->Click();
  ProgHelpForm->Button3->Click();
}
//-----
void __fastcall TForm16::N6Click(TObject *Sender)
```

```

{
    MethHelpForm->Show();
    MethHelpForm->TabSheet3->Show();
}
//-----
void __fastcall TForm16::BitBtn1Click(TObject *Sender)
{int N;
  if (IsNumber(Edit1->Text) && Edit1->Text!="")
  {N=Edit1->Text.ToInt();
    if (N==2)
    {
      X1->Enabled=true;
      X1->Color=clWindow;
      Y1->Enabled=true;
      Y1->Color=clWindow;
      X2->Enabled=true;
      X2->Color=clWindow;
      Y2->Enabled=true;
      Y2->Color=clWindow;
      S1->Enabled=true;
      S1->Color=clWindow;
      S2->Enabled=true;
      S2->Color=clWindow;
      BitBtn1->Caption="Ready";
      TabSheet1->Show();
      X1->SetFocus();
    }
    else if (N==3)
    {
      X1->Enabled=true;
      X1->Color=clWindow;
      Y1->Enabled=true;
      Y1->Color=clWindow;
      X2->Enabled=true;
      X2->Color=clWindow;
      Y2->Enabled=true;
      Y2->Color=clWindow;
      X3->Enabled=true;
      X3->Color=clWindow;
      Y3->Enabled=true;
      Y3->Color=clWindow;
      S1->Enabled=true;
      S1->Color=clWindow;
      S2->Enabled=true;
      S2->Color=clWindow;
      S3->Enabled=true;
      S3->Color=clWindow;
      BitBtn1->Caption="Ready";
      TabSheet1->Show();
      X1->SetFocus();
    }
    else if (N==4)
    {
      X1->Enabled=true;
      X1->Color=clWindow;
      Y1->Enabled=true;
      Y1->Color=clWindow;
      X2->Enabled=true;
      X2->Color=clWindow;
      Y2->Enabled=true;
      Y2->Color=clWindow;
      X3->Enabled=true;
      X3->Color=clWindow;
      Y3->Enabled=true;
      Y3->Color=clWindow;
      X4->Enabled=true;
      X4->Color=clWindow;
      Y4->Enabled=true;
      Y4->Color=clWindow;
      S1->Enabled=true;
      S1->Color=clWindow;
      S2->Enabled=true;
      S2->Color=clWindow;
      S3->Enabled=true;
      S3->Color=clWindow;
      S4->Enabled=true;
      S4->Color=clWindow;
    }
  }
}

```

```

BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==5)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;
S5->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==6)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
X6->Enabled=true;
X6->Color=clWindow;
Y6->Enabled=true;
Y6->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;
S5->Color=clWindow;
S6->Enabled=true;

```



```

S6->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==7)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
X6->Enabled=true;
X6->Color=clWindow;
Y6->Enabled=true;
Y6->Color=clWindow;
X7->Enabled=true;
X7->Color=clWindow;
Y7->Enabled=true;
Y7->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;
S5->Color=clWindow;
S6->Enabled=true;
S6->Color=clWindow;
S7->Enabled=true;
S7->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==8)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
X6->Enabled=true;
X6->Color=clWindow;

```

```

Y6->Enabled=true;
Y6->Color=clWindow;
X7->Enabled=true;
X7->Color=clWindow;
Y7->Enabled=true;
Y7->Color=clWindow;
X8->Enabled=true;
X8->Color=clWindow;
Y8->Enabled=true;
Y8->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;
S5->Color=clWindow;
S6->Enabled=true;
S6->Color=clWindow;
S7->Enabled=true;
S7->Color=clWindow;
S8->Enabled=true;
S8->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==9)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
X6->Enabled=true;
X6->Color=clWindow;
Y6->Enabled=true;
Y6->Color=clWindow;
X7->Enabled=true;
X7->Color=clWindow;
Y7->Enabled=true;
Y7->Color=clWindow;
X8->Enabled=true;
X8->Color=clWindow;
Y8->Enabled=true;
Y8->Color=clWindow;
X9->Enabled=true;
X9->Color=clWindow;
Y9->Enabled=true;
Y9->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;

```

```

S5->Color=clWindow;
S6->Enabled=true;
S6->Color=clWindow;
S7->Enabled=true;
S7->Color=clWindow;
S8->Enabled=true;
S8->Color=clWindow;
S9->Enabled=true;
S9->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
X1->SetFocus();
}
else if (N==10)
{
X1->Enabled=true;
X1->Color=clWindow;
Y1->Enabled=true;
Y1->Color=clWindow;
X2->Enabled=true;
X2->Color=clWindow;
Y2->Enabled=true;
Y2->Color=clWindow;
X3->Enabled=true;
X3->Color=clWindow;
Y3->Enabled=true;
Y3->Color=clWindow;
X4->Enabled=true;
X4->Color=clWindow;
Y4->Enabled=true;
Y4->Color=clWindow;
X5->Enabled=true;
X5->Color=clWindow;
Y5->Enabled=true;
Y5->Color=clWindow;
X6->Enabled=true;
X6->Color=clWindow;
Y6->Enabled=true;
Y6->Color=clWindow;
X7->Enabled=true;
X7->Color=clWindow;
Y7->Enabled=true;
Y7->Color=clWindow;
X8->Enabled=true;
X8->Color=clWindow;
Y8->Enabled=true;
Y8->Color=clWindow;
X9->Enabled=true;
X9->Color=clWindow;
Y9->Enabled=true;
Y9->Color=clWindow;
X10->Enabled=true;
X10->Color=clWindow;
Y10->Enabled=true;
Y10->Color=clWindow;
S1->Enabled=true;
S1->Color=clWindow;
S2->Enabled=true;
S2->Color=clWindow;
S3->Enabled=true;
S3->Color=clWindow;
S4->Enabled=true;
S4->Color=clWindow;
S5->Enabled=true;
S5->Color=clWindow;
S6->Enabled=true;
S6->Color=clWindow;
S7->Enabled=true;
S7->Color=clWindow;
S8->Enabled=true;
S8->Color=clWindow;
S9->Enabled=true;
S9->Color=clWindow;
S10->Enabled=true;
S10->Color=clWindow;
BitBtn1->Caption="Ready";
TabSheet1->Show();
}

```

```

X1->SetFocus();
}
else
    MessageBox(NULL,"Μη Αποδεκτή Τιμή ή Έξω Από Τα Όρια(2 έως 10)!",
        "Σφάλμα",MB_OK|MB_ICONSTOP);
}
else
    MessageBox(NULL,"Μη Αποδεκτός Χαρακτήρας!", "Σφάλμα", MB_OK|
        MB_ICONSTOP);
}
//-----
void __fastcall TForm16::Edit1Change(TObject *Sender)
{
    X1->Enabled=false; X1->Color=clMoneyGreen; X1->Text="";
    Y1->Enabled=false; Y1->Color=clMoneyGreen; Y1->Text="";
    X2->Enabled=false; X2->Color=clMoneyGreen; X2->Text="";
    Y2->Enabled=false; Y2->Color=clMoneyGreen; Y2->Text="";
    X3->Enabled=false; X3->Color=clMoneyGreen; X3->Text="";
    Y3->Enabled=false; Y3->Color=clMoneyGreen; Y3->Text="";
    X4->Enabled=false; X4->Color=clMoneyGreen; X4->Text="";
    Y4->Enabled=false; Y4->Color=clMoneyGreen; Y4->Text="";
    X5->Enabled=false; X5->Color=clMoneyGreen; X5->Text="";
    Y5->Enabled=false; Y5->Color=clMoneyGreen; Y5->Text="";
    X6->Enabled=false; X6->Color=clMoneyGreen; X6->Text="";
    Y6->Enabled=false; Y6->Color=clMoneyGreen; Y6->Text="";
    X7->Enabled=false; X7->Color=clMoneyGreen; X7->Text="";
    Y7->Enabled=false; Y7->Color=clMoneyGreen; Y7->Text="";
    X8->Enabled=false; X8->Color=clMoneyGreen; X8->Text="";
    Y8->Enabled=false; Y8->Color=clMoneyGreen; Y8->Text="";
    X9->Enabled=false; X9->Color=clMoneyGreen; X9->Text="";
    Y9->Enabled=false; Y9->Color=clMoneyGreen; Y9->Text="";
    X10->Enabled=false; X10->Color=clMoneyGreen; X10->Text="";
    Y10->Enabled=false; Y10->Color=clMoneyGreen; Y10->Text="";
    S1->Enabled=false; S1->Color=clMoneyGreen; S1->Text="";
    S2->Enabled=false; S2->Color=clMoneyGreen; S2->Text="";
    S3->Enabled=false; S3->Color=clMoneyGreen; S3->Text="";
    S4->Enabled=false; S4->Color=clMoneyGreen; S4->Text="";
    S5->Enabled=false; S5->Color=clMoneyGreen; S5->Text="";
    S6->Enabled=false; S6->Color=clMoneyGreen; S6->Text="";
    S7->Enabled=false; S7->Color=clMoneyGreen; S7->Text="";
    S8->Enabled=false; S8->Color=clMoneyGreen; S8->Text="";
    S9->Enabled=false; S9->Color=clMoneyGreen; S9->Text="";
    S10->Enabled=false; S10->Color=clMoneyGreen; S10->Text="";
    BitBtn1->Caption="OK";
}
//-----
void __fastcall TForm16::FormHide(TObject *Sender)
{
    Edit1->Text="";
    Edit2->Text="";
    Edit3->Text="";
    Edit4->Text="";
    Edit5->Text="";
}
//-----
void __fastcall TForm16::BitBtn2Click(TObject *Sender)
{
    BOOL EditsReady=false;
    BOOL NotTheSame=false;
    int N=0,i,j;
    double S[10][2], SI[10], sumx=0, sumxq=0, sumy=0, sumxy=0, a, b,
        xab=0, x, sums=0;
    for(i=0;i<10;i++)
    {SI[i]=0;
        for(j=0;j<2;j++)
        {S[i][j]=0;
        }
    }
    if(BitBtn1->Caption=="Ready")
    {N=Edit1->Text.ToInt();
        if(N==2)
        {if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
            Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
            IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(S1->Text) &&
            S1->Text!="" && IsNumber(S2->Text) && S2->Text!="")
        {
            S[0][0]=X1->Text.ToDouble();
            S[0][1]=Y1->Text.ToDouble();
        }
    }
}

```

```

S[1][0]=X2->Text.ToDouble();
S[1][1]=Y2->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0])
    NotTheSame=true;
else
    MessageBox(NULL,"Όμοιες Τιμές X Στα Σημεία, Επίλυση Αδύνατη!","Προσοχή",
        MB_OK|MB_ICONSTOP);
}
}
if(N==3)
{if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
    Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
    IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
    X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
    IsNumber(S1->Text) && S1->Text!="" && IsNumber(S2->Text) &&
    S2->Text!="" && IsNumber(S3->Text) && S3->Text!="")
{
S[0][0]=X1->Text.ToDouble();
S[0][1]=Y1->Text.ToDouble();
S[1][0]=X2->Text.ToDouble();
S[1][1]=Y2->Text.ToDouble();
S[2][0]=X3->Text.ToDouble();
S[2][1]=Y3->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
SI[2]=S3->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[1][0]!=S[2][0])
    NotTheSame=true;
else
    MessageBox(NULL,"Όμοιες Τιμές X Στα Σημεία, Επίλυση Αδύνατη!",
        "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(N==4)
{if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
    Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
    IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
    X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
    IsNumber(X4->Text) && X4->Text!="" && IsNumber(Y4->Text) &&
    Y4->Text!="" && IsNumber(S1->Text) && S1->Text!="" &&
    IsNumber(S2->Text) && S2->Text!="" && IsNumber(S3->Text) &&
    S3->Text!="" && IsNumber(S4->Text) && S4->Text!="")
{
S[0][0]=X1->Text.ToDouble();
S[0][1]=Y1->Text.ToDouble();
S[1][0]=X2->Text.ToDouble();
S[1][1]=Y2->Text.ToDouble();
S[2][0]=X3->Text.ToDouble();
S[2][1]=Y3->Text.ToDouble();
S[3][0]=X4->Text.ToDouble();
S[3][1]=Y4->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
SI[2]=S3->Text.ToDouble();
SI[3]=S4->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
    S[1][0]!=S[2][0] && S[1][0]!=S[3][0] && S[2][0]!=S[3][0])
    NotTheSame=true;
else
    MessageBox(NULL,"Όμοιες Τιμές X Στα Σημεία, Επίλυση Αδύνατη!",
        "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(N==5)
{if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
    Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
    IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
    X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
    IsNumber(X4->Text) && X4->Text!="" && IsNumber(Y4->Text) &&
    Y4->Text!="" && IsNumber(X5->Text) && X5->Text!="" &&
    IsNumber(Y5->Text) && Y5->Text!="" && IsNumber(S1->Text) &&
    S1->Text!="" && IsNumber(S2->Text) && S2->Text!="" &&

```

```

        IsNumber(S3->Text) && S3->Text!="" && IsNumber(S4->Text) &&
        S4->Text!="" && IsNumber(S5->Text) && S5->Text!=""
    {
        S[0][0]=X1->Text.ToDouble();
        S[0][1]=Y1->Text.ToDouble();
        S[1][0]=X2->Text.ToDouble();
        S[1][1]=Y2->Text.ToDouble();
        S[2][0]=X3->Text.ToDouble();
        S[2][1]=Y3->Text.ToDouble();
        S[3][0]=X4->Text.ToDouble();
        S[3][1]=Y4->Text.ToDouble();
        S[4][0]=X5->Text.ToDouble();
        S[4][1]=Y5->Text.ToDouble();
        SI[0]=S1->Text.ToDouble();
        SI[1]=S2->Text.ToDouble();
        SI[2]=S3->Text.ToDouble();
        SI[3]=S4->Text.ToDouble();
        SI[4]=S5->Text.ToDouble();
        EditsReady=true;
        EditsReady=true;
        if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
            S[0][0]!=S[4][0] && S[1][0]!=S[2][0] && S[1][0]!=S[3][0] &&
            S[1][0]!=S[4][0] && S[2][0]!=S[3][0] && S[2][0]!=S[4][0] &&
            S[3][0]!=S[4][0])
            NotTheSame=true;
        else
            MessageBox(NULL,"Όμοιες Τιμές X Στα Σήμεια,Επίλυση Αδύνατη!",
                "Προσοχή", MB_OK|MB_ICONSTOP);
    }
}
if(N==6)
{if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
    Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
    IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
    X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
    IsNumber(X4->Text) && X4->Text!="" && IsNumber(Y4->Text) &&
    Y4->Text!="" && IsNumber(X5->Text) && X5->Text!="" &&
    IsNumber(Y5->Text) && Y5->Text!="" && IsNumber(X6->Text) &&
    X6->Text!="" && IsNumber(Y6->Text) && Y6->Text!="" &&
    IsNumber(X5->Text) && X5->Text!="" && IsNumber(Y5->Text) &&
    Y5->Text!="" && IsNumber(S1->Text) && S1->Text!="" &&
    IsNumber(S2->Text) && S2->Text!="" && IsNumber(S3->Text) &&
    S3->Text!="" && IsNumber(S4->Text) && S4->Text!="" &&
    IsNumber(S5->Text) && S5->Text!="" && IsNumber(S6->Text) &&
    S6->Text!="" )
{
    S[0][0]=X1->Text.ToDouble();
    S[0][1]=Y1->Text.ToDouble();
    S[1][0]=X2->Text.ToDouble();
    S[1][1]=Y2->Text.ToDouble();
    S[2][0]=X3->Text.ToDouble();
    S[2][1]=Y3->Text.ToDouble();
    S[3][0]=X4->Text.ToDouble();
    S[3][1]=Y4->Text.ToDouble();
    S[4][0]=X5->Text.ToDouble();
    S[4][1]=Y5->Text.ToDouble();
    S[5][0]=X6->Text.ToDouble();
    S[5][1]=Y6->Text.ToDouble();
    SI[0]=S1->Text.ToDouble();
    SI[1]=S2->Text.ToDouble();
    SI[2]=S3->Text.ToDouble();
    SI[3]=S4->Text.ToDouble();
    SI[4]=S5->Text.ToDouble();
    SI[5]=S6->Text.ToDouble();
    EditsReady=true;
    if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
        S[0][0]!=S[4][0] && S[0][0]!=S[5][0] && S[1][0]!=S[2][0] &&
        S[1][0]!=S[3][0] && S[1][0]!=S[4][0] && S[1][0]!=S[5][0] &&
        S[2][0]!=S[3][0] && S[2][0]!=S[4][0] && S[2][0]!=S[5][0] &&
        S[3][0]!=S[4][0] && S[3][0]!=S[5][0] && S[4][0]!=S[5][0])
        NotTheSame=true;
    else
        MessageBox(NULL,"Όμοιες Τιμές X Στα Σήμεια,Επίλυση Αδύνατη!",
            "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(N==7)

```

```

{if (IsNumber (X1->Text) && X1->Text!=" && IsNumber (Y1->Text) &&
    Y1->Text!=" && IsNumber (X2->Text) && X2->Text!=" &&
    IsNumber (Y2->Text) && Y2->Text!=" && IsNumber (X3->Text) &&
    X3->Text!=" && IsNumber (Y3->Text) && Y3->Text!=" &&
    IsNumber (X4->Text) && X4->Text!=" && IsNumber (Y4->Text) &&
    Y4->Text!=" && IsNumber (X5->Text) && X5->Text!=" &&
    IsNumber (Y5->Text) && Y5->Text!=" && IsNumber (X6->Text) &&
    X6->Text!=" && IsNumber (Y6->Text) && Y6->Text!=" &&
    IsNumber (X7->Text) && X7->Text!=" && IsNumber (Y7->Text) &&
    Y7->Text!=" && IsNumber (S1->Text) && S1->Text!=" &&
    IsNumber (S2->Text) && S2->Text!=" && IsNumber (S3->Text) &&
    S3->Text!=" && IsNumber (S4->Text) && S4->Text!=" &&
    IsNumber (S5->Text) && S5->Text!=" && IsNumber (S6->Text) &&
    S6->Text!=" && IsNumber (S7->Text) && S7->Text!=")
{
    S[0][0]=X1->Text.ToDouble ();
    S[0][1]=Y1->Text.ToDouble ();
    S[1][0]=X2->Text.ToDouble ();
    S[1][1]=Y2->Text.ToDouble ();
    S[2][0]=X3->Text.ToDouble ();
    S[2][1]=Y3->Text.ToDouble ();
    S[3][0]=X4->Text.ToDouble ();
    S[3][1]=Y4->Text.ToDouble ();
    S[4][0]=X5->Text.ToDouble ();
    S[4][1]=Y5->Text.ToDouble ();
    S[5][0]=X6->Text.ToDouble ();
    S[5][1]=Y6->Text.ToDouble ();
    S[6][0]=X7->Text.ToDouble ();
    S[6][1]=Y7->Text.ToDouble ();
    SI[0]=S1->Text.ToDouble ();
    SI[1]=S2->Text.ToDouble ();
    SI[2]=S3->Text.ToDouble ();
    SI[3]=S4->Text.ToDouble ();
    SI[4]=S5->Text.ToDouble ();
    SI[5]=S6->Text.ToDouble ();
    SI[6]=S7->Text.ToDouble ();
    EditsReady=true;
    if (S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
        S[0][0]!=S[4][0] && S[0][0]!=S[5][0] && S[0][0]!=S[6][0] &&
        S[1][0]!=S[2][0] && S[1][0]!=S[3][0] && S[1][0]!=S[4][0] &&
        S[1][0]!=S[5][0] && S[1][0]!=S[6][0] && S[2][0]!=S[3][0] &&
        S[2][0]!=S[4][0] && S[2][0]!=S[5][0] && S[2][0]!=S[6][0] &&
        S[3][0]!=S[4][0] && S[3][0]!=S[5][0] && S[3][0]!=S[6][0] &&
        S[4][0]!=S[5][0] && S[4][0]!=S[6][0] && S[5][0]!=S[6][0])
        NotTheSame=true;
    else
        MessageBox (NULL, "Όμοιες Τιμές X Στα Σήμεια, Επίλυση Αδύνατη!",
            "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
}
if (N==8)
{if (IsNumber (X1->Text) && X1->Text!=" && IsNumber (Y1->Text) &&
    Y1->Text!=" && IsNumber (X2->Text) && X2->Text!=" &&
    IsNumber (Y2->Text) && Y2->Text!=" && IsNumber (X3->Text) &&
    X3->Text!=" && IsNumber (Y3->Text) && Y3->Text!=" &&
    IsNumber (X4->Text) && X4->Text!=" && IsNumber (Y4->Text) &&
    Y4->Text!=" && IsNumber (X5->Text) && X5->Text!=" &&
    IsNumber (Y5->Text) && Y5->Text!=" && IsNumber (X6->Text) &&
    X6->Text!=" && IsNumber (Y6->Text) && Y6->Text!=" &&
    IsNumber (X7->Text) && X7->Text!=" && IsNumber (Y7->Text) &&
    Y7->Text!=" && IsNumber (X8->Text) && X8->Text!=" &&
    IsNumber (Y8->Text) && Y8->Text!=" && IsNumber (S1->Text) &&
    S1->Text!=" && IsNumber (S2->Text) && S2->Text!=" &&
    IsNumber (S3->Text) && S3->Text!=" && IsNumber (S4->Text) &&
    S4->Text!=" && IsNumber (S5->Text) && S5->Text!=" &&
    IsNumber (S6->Text) && S6->Text!=" && IsNumber (S7->Text) &&
    S7->Text!=" && IsNumber (S8->Text) && S8->Text!=")
{
    S[0][0]=X1->Text.ToDouble ();
    S[0][1]=Y1->Text.ToDouble ();
    S[1][0]=X2->Text.ToDouble ();
    S[1][1]=Y2->Text.ToDouble ();
    S[2][0]=X3->Text.ToDouble ();
    S[2][1]=Y3->Text.ToDouble ();
    S[3][0]=X4->Text.ToDouble ();
    S[3][1]=Y4->Text.ToDouble ();
    S[4][0]=X5->Text.ToDouble ();

```

```

S[4][1]=Y5->Text.ToDouble();
S[5][0]=X6->Text.ToDouble();
S[5][1]=Y6->Text.ToDouble();
S[6][0]=X7->Text.ToDouble();
S[6][1]=Y7->Text.ToDouble();
S[7][0]=X8->Text.ToDouble();
S[7][1]=Y8->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
SI[2]=S3->Text.ToDouble();
SI[3]=S4->Text.ToDouble();
SI[4]=S5->Text.ToDouble();
SI[5]=S6->Text.ToDouble();
SI[6]=S7->Text.ToDouble();
SI[7]=S8->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
S[0][0]!=S[4][0] && S[0][0]!=S[5][0] && S[0][0]!=S[6][0] &&
S[0][0]!=S[7][0] && S[1][0]!=S[2][0] && S[1][0]!=S[3][0] &&
S[1][0]!=S[4][0] && S[1][0]!=S[5][0] && S[1][0]!=S[6][0] &&
S[1][0]!=S[7][0] && S[2][0]!=S[3][0] && S[2][0]!=S[4][0] &&
S[2][0]!=S[5][0] && S[2][0]!=S[6][0] && S[2][0]!=S[7][0] &&
S[3][0]!=S[4][0] && S[3][0]!=S[5][0] && S[3][0]!=S[6][0] &&
S[3][0]!=S[7][0] && S[4][0]!=S[5][0] && S[4][0]!=S[6][0] &&
S[4][0]!=S[7][0] && S[5][0]!=S[6][0] &&
S[5][0]!=S[7][0] && S[6][0]!=S[7][0])
    NotTheSame=true;
else
    MessageBox(NULL,"Όμοιες Τιμές Χ Στα Σήμεια,Επίλυση Αδύνατη!",
        "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(N==9)
{if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
IsNumber(X4->Text) && X4->Text!="" && IsNumber(Y4->Text) &&
Y4->Text!="" && IsNumber(X5->Text) && X5->Text!="" &&
IsNumber(Y5->Text) && Y5->Text!="" && IsNumber(X6->Text) &&
X6->Text!="" && IsNumber(Y6->Text) && Y6->Text!="" &&
IsNumber(X7->Text) && X7->Text!="" && IsNumber(Y7->Text) &&
Y7->Text!="" && IsNumber(X8->Text) && X8->Text!="" &&
IsNumber(Y8->Text) && Y8->Text!="" && IsNumber(X9->Text) &&
X9->Text!="" && IsNumber(Y9->Text) && Y9->Text!="" &&
IsNumber(S1->Text) && S1->Text!="" && IsNumber(S2->Text) &&
S2->Text!="" && IsNumber(S3->Text) && S3->Text!="" &&
IsNumber(S4->Text) && S4->Text!="" && IsNumber(S5->Text) &&
S5->Text!="" && IsNumber(S6->Text) && S6->Text!="" &&
IsNumber(S7->Text) && S7->Text!="" && IsNumber(S8->Text) &&
S8->Text!="" && IsNumber(S9->Text) && S9->Text!="")
{
S[0][0]=X1->Text.ToDouble();
S[0][1]=Y1->Text.ToDouble();
S[1][0]=X2->Text.ToDouble();
S[1][1]=Y2->Text.ToDouble();
S[2][0]=X3->Text.ToDouble();
S[2][1]=Y3->Text.ToDouble();
S[3][0]=X4->Text.ToDouble();
S[3][1]=Y4->Text.ToDouble();
S[4][0]=X5->Text.ToDouble();
S[4][1]=Y5->Text.ToDouble();
S[5][0]=X6->Text.ToDouble();
S[5][1]=Y6->Text.ToDouble();
S[6][0]=X7->Text.ToDouble();
S[6][1]=Y7->Text.ToDouble();
S[7][0]=X8->Text.ToDouble();
S[7][1]=Y8->Text.ToDouble();
S[8][0]=X9->Text.ToDouble();
S[8][1]=Y9->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
SI[2]=S3->Text.ToDouble();
SI[3]=S4->Text.ToDouble();
SI[4]=S5->Text.ToDouble();
SI[5]=S6->Text.ToDouble();
SI[6]=S7->Text.ToDouble();
}
}

```



```

SI[7]=S8->Text.ToDouble();
SI[8]=S9->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
    S[0][0]!=S[4][0] && S[0][0]!=S[5][0] && S[0][0]!=S[6][0] &&
    S[0][0]!=S[7][0] && S[0][0]!=S[8][0] && S[1][0]!=S[2][0] &&
    S[1][0]!=S[3][0] && S[1][0]!=S[4][0] && S[1][0]!=S[5][0] &&
    S[1][0]!=S[6][0] && S[1][0]!=S[7][0] && S[1][0]!=S[8][0] &&
    S[2][0]!=S[3][0] && S[2][0]!=S[4][0] && S[2][0]!=S[5][0] &&
    S[2][0]!=S[6][0] && S[2][0]!=S[7][0] && S[2][0]!=S[8][0] &&
    S[3][0]!=S[4][0] && S[3][0]!=S[5][0] && S[3][0]!=S[6][0] &&
    S[3][0]!=S[7][0] && S[3][0]!=S[8][0] && S[4][0]!=S[5][0] &&
    S[4][0]!=S[6][0] && S[4][0]!=S[7][0] && S[4][0]!=S[8][0] &&
    S[5][0]!=S[6][0] && S[5][0]!=S[7][0] && S[5][0]!=S[8][0] &&
    S[6][0]!=S[7][0] && S[6][0]!=S[8][0] && S[7][0]!=S[8][0])
    NotTheSame=true;
else
    MessageBox(NULL,"Ομοιες Τιμές Χ Στα Σήμεια,Επίλυση Αδύνατη!",
        "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(N==10)
if(IsNumber(X1->Text) && X1->Text!="" && IsNumber(Y1->Text) &&
    Y1->Text!="" && IsNumber(X2->Text) && X2->Text!="" &&
    IsNumber(Y2->Text) && Y2->Text!="" && IsNumber(X3->Text) &&
    X3->Text!="" && IsNumber(Y3->Text) && Y3->Text!="" &&
    IsNumber(X4->Text) && X4->Text!="" && IsNumber(Y4->Text) &&
    Y4->Text!="" && IsNumber(X5->Text) && X5->Text!="" &&
    IsNumber(Y5->Text) && Y5->Text!="" && IsNumber(X6->Text) &&
    X6->Text!="" && IsNumber(Y6->Text) && Y6->Text!="" &&
    IsNumber(X7->Text) && X7->Text!="" && IsNumber(Y7->Text) &&
    Y7->Text!="" && IsNumber(X8->Text) && X8->Text!="" &&
    IsNumber(Y8->Text) && Y8->Text!="" && IsNumber(X9->Text) &&
    X9->Text!="" && IsNumber(Y9->Text) && Y9->Text!="" &&
    IsNumber(X10->Text) && X10->Text!="" && IsNumber(Y10->Text) &&
    Y10->Text!="" && IsNumber(S1->Text) && S1->Text!="" &&
    IsNumber(S2->Text) && S2->Text!="" && IsNumber(S3->Text) &&
    S3->Text!="" && IsNumber(S4->Text) && S4->Text!="" &&
    IsNumber(S5->Text) && S5->Text!="" && IsNumber(S6->Text) &&
    S6->Text!="" && IsNumber(S7->Text) && S7->Text!="" &&
    IsNumber(S8->Text) && S8->Text!="" && IsNumber(S9->Text) &&
    S9->Text!="" && IsNumber(S10->Text) && S10->Text!="")
{
S[0][0]=X1->Text.ToDouble();
S[0][1]=Y1->Text.ToDouble();
S[1][0]=X2->Text.ToDouble();
S[1][1]=Y2->Text.ToDouble();
S[2][0]=X3->Text.ToDouble();
S[2][1]=Y3->Text.ToDouble();
S[3][0]=X4->Text.ToDouble();
S[3][1]=Y4->Text.ToDouble();
S[4][0]=X5->Text.ToDouble();
S[4][1]=Y5->Text.ToDouble();
S[5][0]=X6->Text.ToDouble();
S[5][1]=Y6->Text.ToDouble();
S[6][0]=X7->Text.ToDouble();
S[6][1]=Y7->Text.ToDouble();
S[7][0]=X8->Text.ToDouble();
S[7][1]=Y8->Text.ToDouble();
S[8][0]=X9->Text.ToDouble();
S[8][1]=Y9->Text.ToDouble();
S[9][0]=X10->Text.ToDouble();
S[9][1]=Y10->Text.ToDouble();
SI[0]=S1->Text.ToDouble();
SI[1]=S2->Text.ToDouble();
SI[2]=S3->Text.ToDouble();
SI[3]=S4->Text.ToDouble();
SI[4]=S5->Text.ToDouble();
SI[5]=S6->Text.ToDouble();
SI[6]=S7->Text.ToDouble();
SI[7]=S8->Text.ToDouble();
SI[8]=S9->Text.ToDouble();
SI[9]=S10->Text.ToDouble();
EditsReady=true;
if(S[0][0]!=S[1][0] && S[0][0]!=S[2][0] && S[0][0]!=S[3][0] &&
    S[0][0]!=S[4][0] && S[0][0]!=S[5][0] && S[0][0]!=S[6][0] &&
    S[0][0]!=S[7][0] && S[0][0]!=S[8][0] && S[0][0]!=S[9][0] &&

```

```

S[1][0]!=S[2][0] && S[1][0]!=S[3][0] && S[1][0]!=S[4][0] &&
S[1][0]!=S[5][0] && S[1][0]!=S[6][0] && S[1][0]!=S[7][0] &&
S[1][0]!=S[8][0] && S[1][0]!=S[9][0] && S[2][0]!=S[3][0] &&
S[2][0]!=S[4][0] && S[2][0]!=S[5][0] && S[2][0]!=S[6][0] &&
S[2][0]!=S[7][0] && S[2][0]!=S[8][0] && S[2][0]!=S[9][0] &&
S[3][0]!=S[4][0] && S[3][0]!=S[5][0] && S[3][0]!=S[6][0] &&
S[3][0]!=S[7][0] && S[3][0]!=S[8][0] && S[3][0]!=S[9][0] &&
S[4][0]!=S[5][0] && S[4][0]!=S[6][0] && S[4][0]!=S[7][0] &&
S[4][0]!=S[8][0] && S[4][0]!=S[9][0] && S[5][0]!=S[6][0] &&
S[5][0]!=S[7][0] && S[5][0]!=S[8][0] && S[5][0]!=S[9][0] &&
S[6][0]!=S[7][0] && S[6][0]!=S[8][0] && S[7][0]!=S[8][0] &&
S[7][0]!=S[9][0] && S[8][0]!=S[9][0])
NotTheSame=true;
else
    MessageBox(NULL,"Όμοιες Τιμές X Στα Σήμεια,Επίλυση Αδύνατη!",
        "Προσοχή", MB_OK|MB_ICONSTOP);
}
}
if(EditsReady && NotTheSame)
{
    for(i=0;i<N;i++)
        sums=sumx+(1/(SI[i]*SI[i]));
    for(i=0;i<N;i++)
        sumx=sumx+(S[i][0]/(SI[i]*SI[i]));
    for(i=0;i<N;i++)
        sumy=sumy+(S[i][1]/(SI[i]*SI[i]));
    for(i=0;i<N;i++)
        sumxq=sumxq+((S[i][0]*S[i][0])/(SI[i]*SI[i]));
    for(i=0;i<N;i++)
        sumxy=sumxy+((S[i][0]*S[i][1])/(SI[i]*SI[i]));
    a=((sumy*sumxq)-(sumx*sumxy))/((sums*sumxq)-(sumx*sumx));
    b=((sums*sumy)-(sumx*sumy))/((sums*sumxq)-(sumx*sumx));
    Edit2->Text=a;
    Edit3->Text=b;
    for(i=0;i<N;i++)
    {
        xab=xab+((S[i][1]-a-b*S[i][0])/SI[i])*((S[i][1]-a-b*S[i][0])/SI[i]);
    }
    Edit4->Text=xab;
    x=-a/b;
    Edit5->Text=x;
    Button1->Click();
}
else
    MessageBox(NULL,"Δεν Υπάρχουν Τιμές Σημείων!", "Σφάλμα", MB_OK|
        MB_ICONSTOP);
}
else
    MessageBox(NULL,"Δεν Ορίστηκε Αριθμός Σημείων!", "Σφάλμα", MB_OK|
        MB_ICONSTOP);
}
//-----
void __fastcall TForm16::Button1Click(TObject *Sender)
{
    int N,Height,Width,X0,Y0,xg,yg;
    double xgr,ygr,xlgr,ylgr,x2gr,y2gr,XScale,YScale,xstep;
    double x,Ed1,Ed2,Ed3,Ed4;
    double X11, X22, X33, X44, X55, X66, X77, X88, X99, X100, Y11, Y22,
        Y33, Y44, Y55, Y66, Y77, Y88, Y99, Y100;

    if(IsNumber(Edit1->Text) && Edit1->Text!="")
        N=Edit1->Text.ToInt();

    if(N==2&&X1->Text!=""&&X2->Text!="" && Y1->Text!="" && Y2->Text!="")
        {X11=X1->Text.ToDouble();X22=X2->Text.ToDouble();
        Y11=Y1->Text.ToDouble();Y22=Y2->Text.ToDouble();}
    if(Edit5->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);

        Label33->Visible=false;Label34->Visible=false;Label39->Visible=false;
        Label40->Visible=false;Label37->Visible=false;Label38->Visible=false;
        Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
        Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
    }
}

```

```

Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

Shape2->Visible=false;Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
}

else
{
    Ed1=Min(X11,X22);
    Ed2=Max(X11,X22);
    Ed3=Min(Y11,Y22);
    Ed4=Max(Y11,Y22);

    Image1->Visible=true;
    Shape1->Visible=true;

Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

    Label132->Caption=Max(fabs(Ed1), fabs(Ed2));
    Label124->Caption=(Label132->Caption)*(-1.0);
    Label130->Caption=Label132->Caption.ToDouble()/4.0;
    Label131->Caption=Label132->Caption.ToDouble()/2.0;
    Label128->Caption=(Label132->Caption.ToDouble()/4.0)*3.0;
    Label125->Caption=(Label128->Caption)*(-1.0);
    Label126->Caption=(Label131->Caption)*(-1.0);
    Label127->Caption=(Label130->Caption)*(-1.0);

    Label133->Caption=Max(fabs(Ed3), fabs(Ed4));
    Label138->Caption=(Label133->Caption)*(-1.0);
    Label134->Caption=(Label133->Caption.ToDouble()/3.0)*2.0;
    Label137->Caption=(Label133->Caption.ToDouble()/3.0);
    Label139->Caption=(Label134->Caption.ToDouble())*(-1.0);
    Label140->Caption=(Label137->Caption.ToDouble())*(-1.0);

if(Ed1<Ed2)//X1->Text<X2->Text)
{
    if(Edit5->Text=="")
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα","Προσοχή", MB_OK|
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;
        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Ed1;
        x2gr=Ed2;
        y1gr=Ed3;
        y2gr=Ed4;
    }
}

```

```

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=func(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg,yg);
}
}

else if (Ed1>Ed2) //X1->Text>X2->Text
{
if (Edit5->Text=="")
MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
MB_ICONINFORMATION);
else
{
xgr=Edit5->Text.ToDouble();

Height=Image1->Picture->Height;
Width=Image1->Picture->Width;

X0=Width/2;
Y0=Height/2;

Image1->Canvas->Brush->Color=ImageCanBrush;
Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect(Rect(0,0,Width,Height));

Image1->Canvas->Pen->Color=ImageCanPen;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed2;
x2gr=Ed1;
y1gr=Ed4;
y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=func(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg,yg);
}
} //megalos else
Point1->Top=Image1->Top+(Image1->Height)/2+
((Y1->Text.ToDouble())*(-41))*(3/(Label33->Caption.ToDouble()))-1;
Point1->Left=Image1->Left+(Image1->Width)/2+
((X1->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

```

```

    Point2->Top=Imagel->Top + (Imagel->Height)/2 +
    ((Y2->Text.ToDouble())*(-41))*(3/(Label33->Caption.ToDouble()))-1;
    Point2->Left=Imagel->Left + (Imagel->Width)/2 +
    ((X2->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point1->Visible=true;Point2->Visible=true;
Point3->Visible=false; Point4->Visible=false; Point5->Visible=false;
Point6->Visible=false; Point7->Visible=false; Point8->Visible=false;
Point9->Visible=false; Point10->Visible=false;

} //N==2

if(N==3 && X1->Text!=""&& X2->Text!=""&& X3->Text!="" &&
    Y1->Text!=""&& Y2->Text!=""&& Y3->Text!="")
{
    X11=X1->Text.ToDouble();X22=X2->Text.ToDouble();
    X33=X3->Text.ToDouble();Y11=Y1->Text.ToDouble();
    Y22=Y2->Text.ToDouble();Y33=Y3->Text.ToDouble();

    if(Edit5->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);

Label133->Visible=false;Label134->Visible=false;Label139->Visible=false;
Label140->Visible=false;Label137->Visible=false;Label138->Visible=false;
Label124->Visible=false;Label125->Visible=false;Label126->Visible=false;
Label127->Visible=false;Label128->Visible=false;Label129->Visible=false;
Label130->Visible=false;Label131->Visible=false;Label132->Visible=false;

Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }

    else
    {
        Ed1=Min(X11,Min(X22,X33));
        Ed2=Max(X11,Max(X22,X33));
        Ed3=Min(Y11,Min(Y22,Y33));
        Ed4=Max(Y11,Max(Y22,Y33));

        Imagel->Visible=true;
        Shape1->Visible=true;

Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

        Label32->Caption=Max(fabs(Ed1), fabs(Ed2));
        Label124->Caption=(Label32->Caption)*(-1.0);
        Label130->Caption=Label32->Caption.ToDouble()/4.0;
        Label131->Caption=Label32->Caption.ToDouble()/2.0;
        Label128->Caption=(Label32->Caption.ToDouble()/4.0)*3.0;
        Label125->Caption=(Label128->Caption)*(-1.0);
        Label126->Caption=(Label131->Caption)*(-1.0);
        Label127->Caption=(Label130->Caption)*(-1.0);

        Label133->Caption=Max(fabs(Ed3), fabs(Ed4));
        Label138->Caption=(Label133->Caption)*(-1.0);
        Label134->Caption=(Label133->Caption.ToDouble()/3.0)*2.0;
        Label137->Caption=(Label133->Caption.ToDouble()/3.0);
        Label139->Caption=(Label134->Caption.ToDouble())*(-1.0);
        Label140->Caption=(Label137->Caption.ToDouble())*(-1.0);

    if(Ed1<Ed2)
    {

```

```

if (Edit5->Text=="")
    MessageBox (NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect (Rect (0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo (X0,0);
    Image1->Canvas->LineTo (X0,Height);
    Image1->Canvas->MoveTo (0,Y0);
    Image1->Canvas->LineTo (Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Ed1;
    x2gr=Ed2;
    y1gr=Ed3;
    y2gr=Ed4;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max (fabs (x1gr), fabs (x2gr));
    YScale=Height/2/Max (fabs (y1gr), fabs (y2gr));
} //else

    ygr=func (x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo (xg, yg);

    for (xgr=x1gr; xgr<=x2gr; xgr+=xstep)
    {
        ygr=func (xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo (xg, yg);
    }
}

else if (Ed1>Ed2) //X1->Text>X2->Text)
{
    if (Edit5->Text=="")
        MessageBox (NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                    MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect (Rect (0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo (X0,0);
        Image1->Canvas->LineTo (X0,Height);
        Image1->Canvas->MoveTo (0,Y0);
        Image1->Canvas->LineTo (Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;
    }
}

```

```

x1gr=Ed2;
x2gr=Ed1;
y1gr=Ed4;
y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Imagel->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
    ygr=func(xgr);
    xg=xgr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Imagel->Canvas->LineTo(xg,yg);
}
} //megalo else
Point1->Top=Imagel->Top + (Imagel->Height)/2 +
((Y1->Text.ToDouble())*(-41))*(3/(Label133->Caption.ToDouble()))-1;
Point1->Left=Imagel->Left + (Imagel->Width)/2 +
((X1->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point2->Top=Imagel->Top + (Imagel->Height)/2 +
((Y2->Text.ToDouble())*(-41))*(3/(Label133->Caption.ToDouble()))-1;
Point2->Left=Imagel->Left + (Imagel->Width)/2 +
((X2->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point3->Top=Imagel->Top + (Imagel->Height)/2 +
((Y3->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point3->Left=Imagel->Left + (Imagel->Width)/2 +
((X3->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=false; Point5->Visible=false; Point6->Visible=false;
Point7->Visible=false; Point8->Visible=false; Point9->Visible=false;
Point10->Visible=false;

} //N==3

if(N==4 && X1->Text!=""&& X2->Text!=""&& X3->Text!=""&& X4->Text!=""
&& Y1->Text!="" && Y2->Text!=""&& Y3->Text!=""&& Y4->Text!="")
{
    X11=X1->Text.ToDouble();X22=X2->Text.ToDouble();
    X33=X3->Text.ToDouble();X44=X4->Text.ToDouble();
    Y11=Y1->Text.ToDouble();Y22=Y2->Text.ToDouble();
    Y33=Y3->Text.ToDouble();Y44=Y4->Text.ToDouble();

    if(Edit5->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
        MB_ICONINFORMATION);

Label133->Visible=false;Label134->Visible=false;Label139->Visible=false;
Label140->Visible=false;Label137->Visible=false;Label138->Visible=false;
Label124->Visible=false;Label125->Visible=false;Label126->Visible=false;
Label127->Visible=false;Label128->Visible=false;Label129->Visible=false;
Label130->Visible=false;Label131->Visible=false;Label132->Visible=false;

Shape2->Visible=false;Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }

else
{
    Ed1=Min(X11,Min(X22,Min(X33,X44)));
}

```

```

Ed2=Max (X11,Max (X22,Max (X33,X44) ) ) ;
Ed3=Min (Y11,Min (Y22,Min (Y33,Y44) ) ) ;
Ed4=Max (Y11,Max (Y22,Max (Y33,Y44) ) ) ;

Image1->Visible=true;
Shape1->Visible=true;

Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

Label132->Caption=Max (fabs (Ed1) , fabs (Ed2) ) ;
Label124->Caption=(Label132->Caption) * (-1.0) ;
Label130->Caption=Label132->Caption.ToDouble () /4.0 ;
Label131->Caption=Label132->Caption.ToDouble () /2.0 ;
Label128->Caption=(Label132->Caption.ToDouble () /4.0) *3.0 ;
Label125->Caption=(Label128->Caption) * (-1.0) ;
Label126->Caption=(Label131->Caption) * (-1.0) ;
Label127->Caption=(Label130->Caption) * (-1.0) ;

Label133->Caption=Max (fabs (Ed3) , fabs (Ed4) ) ;
Label138->Caption=(Label133->Caption) * (-1.0) ;
Label134->Caption=(Label133->Caption.ToDouble () /3.0) *2.0 ;
Label137->Caption=(Label133->Caption.ToDouble () ) /3.0 ;
Label139->Caption=(Label134->Caption.ToDouble () ) * (-1.0) ;
Label140->Caption=(Label137->Caption.ToDouble () ) * (-1.0) ;

if (Ed1<Ed2)
{
if (Edit5->Text=="")
MessageBox (NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
MB_ICONINFORMATION) ;
else
{
xgr=Edit5->Text.ToDouble () ;

Height=Image1->Picture->Height ;
Width=Image1->Picture->Width ;

X0=Width/2 ;
Y0=Height/2 ;

Image1->Canvas->Brush->Color=ImageCanBrush ;
Image1->Canvas->Brush->Style=bsSolid ;
Image1->Canvas->FillRect (Rect (0,0,Width,Height) ) ;

Image1->Canvas->Pen->Color=ImageCanPen ;
Image1->Canvas->Pen->Width=2 ;
Image1->Canvas->MoveTo (X0,0) ;
Image1->Canvas->LineTo (X0,Height) ;
Image1->Canvas->MoveTo (0,Y0) ;
Image1->Canvas->LineTo (Width,Y0) ;
Image1->Canvas->Pen->Color=LineColor ;

x1gr=Ed1 ;
x2gr=Ed2 ;
y1gr=Ed3 ;
y2gr=Ed4 ;

xstep=(x2gr-x1gr) /100.0 ;

XScale=Width/2/Max (fabs (x1gr) , fabs (x2gr) ) ;
YScale=Height/2/Max (fabs (y1gr) , fabs (y2gr) ) ;
} //else

ygr=func (x1gr) ;
xg=x1gr*XScale+X0 ;
yg=-ygr*YScale+Y0 ;
Image1->Canvas->MoveTo (xg,yg) ;

```



```

        for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
        {
            ygr=func(xgr);
            xg=xgr*XScale+X0;
            yg=-ygr*YScale+Y0;
            Image1->Canvas->LineTo(xg,yg);
        }
    }

else if (Ed1>Ed2)
{
    if (Edit5->Text=="")
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Ed2;
        x2gr=Ed1;
        y1gr=Ed4;
        y2gr=Ed3;
        xstep=(x2gr-x1gr)/100.0;

        XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
        YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
    }//else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

//megalo else
Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble())*(-41))* (3/(Label33->Caption.ToDouble()))-1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble())*46)* (3.9/(Label32->Caption.ToDouble()))-1;

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble())*(-41))* (3/(Label33->Caption.ToDouble()))-1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble())*46)* (3.9/(Label32->Caption.ToDouble()))-1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble())*(-41))* (2.9/(Label33->Caption.ToDouble()))-1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble())*46)* (3.9/(Label32->Caption.ToDouble()))-1;

Point4->Top=Image1->Top + (Image1->Height)/2 +

```

```

((Y4->Text.ToDouble()) * (-41)) * (2.9 / (Label33->Caption.ToDouble())) - 1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble()) * 46) * (3.9 / (Label32->Caption.ToDouble())) - 1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=false; Point6->Visible=false;
Point7->Visible=false; Point8->Visible=false; Point9->Visible=false;
Point10->Visible=false;
} //N==4

if (N==5 && X1->Text!=""&& X2->Text!=""&& X3->Text!=""&&
X4->Text!=""&& X5->Text!=""&& Y1->Text!=""&&
Y2->Text!=""&& Y3->Text!=""&& Y4->Text!=""&& Y5->Text!="")
{
X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
X55=X5->Text.ToDouble(); Y11=Y1->Text.ToDouble();
Y22=Y2->Text.ToDouble(); Y33=Y3->Text.ToDouble();
Y44=Y4->Text.ToDouble(); Y55=Y5->Text.ToDouble();

if (Edit5->Text=="")
{
MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
MB_ICONINFORMATION);

Label133->Visible=false; Label134->Visible=false; Label139->Visible=false;
Label140->Visible=false; Label137->Visible=false; Label138->Visible=false;
Label124->Visible=false; Label125->Visible=false; Label126->Visible=false;
Label127->Visible=false; Label128->Visible=false; Label129->Visible=false;
Label130->Visible=false; Label131->Visible=false; Label132->Visible=false;

Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
}

else
{
Ed1=Min(X11, Min(X22, Min(X33, Min(X44, X55))));
Ed2=Max(X11, Max(X22, Max(X33, Max(X44, X55))));
Ed3=Min(Y11, Min(Y22, Min(Y33, Min(Y44, Y55))));
Ed4=Max(Y11, Max(Y22, Max(Y33, Max(Y44, Y55))));

Image1->Visible=true;
Shape1->Visible=true;
Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

Label32->Caption=Max(fabs(Ed1), fabs(Ed2));
Label24->Caption=(Label32->Caption) * (-1.0);
Label30->Caption=Label32->Caption.ToDouble() / 4.0;
Label31->Caption=Label32->Caption.ToDouble() / 2.0;
Label28->Caption=(Label32->Caption.ToDouble() / 4.0) * 3.0;
Label25->Caption=(Label28->Caption) * (-1.0);
Label26->Caption=(Label31->Caption) * (-1.0);
Label27->Caption=(Label30->Caption) * (-1.0);

Label33->Caption=Max(fabs(Ed3), fabs(Ed4));
Label38->Caption=(Label33->Caption) * (-1.0);
Label34->Caption=(Label33->Caption.ToDouble() / 3.0) * 2.0;
Label37->Caption=(Label33->Caption.ToDouble() / 3.0);
Label39->Caption=(Label34->Caption.ToDouble() * (-1.0));
Label40->Caption=(Label37->Caption.ToDouble() * (-1.0));

if (Ed1 < Ed2)
{

```

```

if (Edit5->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Ed1;
    x2gr=Ed2;
    y1gr=Ed3;
    y2gr=Ed4;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

else if (Ed1>Ed2)
{
    if (Edit5->Text=="")
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                    MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;
    }
}

```

```

x1gr=Ed2;
x2gr=Ed1;
y1gr=Ed4;
y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Imagel->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
ygr=func(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Imagel->Canvas->LineTo(xg,yg);
}
} //megalo else
Point1->Top=Imagel->Top + (Imagel->Height)/2 +
((Y1->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point1->Left=Imagel->Left + (Imagel->Width)/2 +
((X1->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point2->Top=Imagel->Top + (Imagel->Height)/2 +
((Y2->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point2->Left=Imagel->Left + (Imagel->Width)/2 +
((X2->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point3->Top=Imagel->Top + (Imagel->Height)/2 +
((Y3->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point3->Left=Imagel->Left + (Imagel->Width)/2 +
((X3->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point4->Top=Imagel->Top + (Imagel->Height)/2 +
((Y4->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point4->Left=Imagel->Left + (Imagel->Width)/2 +
((X4->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point5->Top=Imagel->Top + (Imagel->Height)/2 +
((Y5->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point5->Left=Imagel->Left + (Imagel->Width)/2 +
((X5->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=false;
Point7->Visible=false; Point8->Visible=false; Point9->Visible=false;
Point10->Visible=false;

} //N==5

if(N==6 && X1->Text!="" && X2->Text!="" && X3->Text!="" &&
X4->Text!="" && X5->Text!="" && X6->Text!="" &&
Y1->Text!="" && Y2->Text!="" && Y3->Text!="" &&
Y4->Text!="" && Y5->Text!="" && Y6->Text!="")
{
X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
X55=X5->Text.ToDouble(); X66=X6->Text.ToDouble();
Y11=Y1->Text.ToDouble(); Y22=Y2->Text.ToDouble();
Y33=Y3->Text.ToDouble(); Y44=Y4->Text.ToDouble();
Y55=Y5->Text.ToDouble(); Y66=Y6->Text.ToDouble();

if(Edit5->Text=="")
{
MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή" ,MB_OK|
MB_ICONINFORMATION);
Label133->Visible=false;Label134->Visible=false;Label139->Visible=false;

```

```

Label40->Visible=false;Label37->Visible=false;Label38->Visible=false;
Label24->Visible=false;Label25->Visible=false;Label26->Visible=false;
Label27->Visible=false;Label28->Visible=false;Label29->Visible=false;
Label30->Visible=false;Label31->Visible=false;Label32->Visible=false;

Shape2->Visible=false;Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false;Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
}

else
{
    Ed1=Min (X11,Min (X22,Min (X33,Min (X44,Min (X55,X66) ) ) ) ) );
    Ed2=Max (X11,Max (X22,Max (X33,Max (X44,Max (X55,X66) ) ) ) ) );
    Ed3=Min (Y11,Min (Y22,Min (Y33,Min (Y44,Min (Y55,Y66) ) ) ) ) );
    Ed4=Max (Y11,Max (Y22,Max (Y33,Max (Y44,Max (Y55,Y66) ) ) ) ) );

    Image1->Visible=true;
    Shape1->Visible=true;
Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label33->Visible=true; Label34->Visible=true; Label39->Visible=true;
Label40->Visible=true; Label37->Visible=true; Label38->Visible=true;
Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

    Label32->Caption=Max (fabs (Ed1) , fabs (Ed2) ) );
    Label24->Caption=(Label32->Caption) * (-1.0);
    Label30->Caption=Label32->Caption.ToDouble () / 4.0;
    Label31->Caption=Label32->Caption.ToDouble () / 2.0;
    Label28->Caption=(Label32->Caption.ToDouble () / 4.0) * 3.0;
    Label25->Caption=(Label28->Caption) * (-1.0);
    Label26->Caption=(Label31->Caption) * (-1.0);
    Label27->Caption=(Label30->Caption) * (-1.0);

    Label33->Caption=Max (fabs (Ed3) , fabs (Ed4) ) );
    Label38->Caption=(Label33->Caption) * (-1.0);
    Label34->Caption=(Label33->Caption.ToDouble () / 3.0) * 2.0;
    Label37->Caption=(Label33->Caption.ToDouble () ) / 3.0;
    Label39->Caption=(Label34->Caption.ToDouble () ) * (-1.0);
    Label40->Caption=(Label37->Caption.ToDouble () ) * (-1.0);

if (Ed1<Ed2)
{
    if (Edit5->Text=="")
        MessageBox (NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble ();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect (Rect (0,0,Width,Height) );

        Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo (X0,0);
        Image1->Canvas->LineTo (X0,Height);
        Image1->Canvas->MoveTo (0,Y0);
        Image1->Canvas->LineTo (Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Ed1;

```

```

x2gr=Ed2;
y1gr=Ed3;
y2gr=Ed4;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg, yg);

for(xgr=x1gr; xgr<=x2gr; xgr+=xstep)
{
ygr=func(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg, yg);
}
}

else if(Ed1>Ed2) //X1->Text>X2->Text
{
if(Edit5->Text=="")
MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
MB_ICONINFORMATION);
else
{
xgr=Edit5->Text.ToDouble();

Height=Image1->Picture->Height;
Width=Image1->Picture->Width;

X0=Width/2;
Y0=Height/2;

Image1->Canvas->Brush->Color=ImageCanBrush; //clNavy;
Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect(Rect(0, 0, Width, Height));

Image1->Canvas->Pen->Color=ImageCanPen; //clMaroon;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo(X0, 0);
Image1->Canvas->LineTo(X0, Height);
Image1->Canvas->MoveTo(0, Y0);
Image1->Canvas->LineTo(Width, Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed2; x2gr=Ed1; y1gr=Ed4; y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg, yg);

for(xgr=x1gr; xgr<=x2gr; xgr+=xstep)
{
ygr=func(xgr);
xg=xgr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->LineTo(xg, yg);
}
} //megalos else
Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble())*(-41))* (3/(Label33->Caption.ToDouble()))-1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble())*46)* (3.9/(Label32->Caption.ToDouble()))-1;

```

```

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble())*(-41))*(3/(Label33->Caption.ToDouble()))-1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point4->Top=Image1->Top + (Image1->Height)/2 +
((Y4->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point5->Top=Image1->Top + (Image1->Height)/2 +
((Y5->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
Point5->Left=Image1->Left + (Image1->Width)/2 +
((X5->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point6->Top=Image1->Top + (Image1->Height)/2 +
((Y6->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
Point6->Left=Image1->Left + (Image1->Width)/2 +
((X6->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=false; Point8->Visible=false; Point9->Visible=false;
Point10->Visible=false;

} //N==6

if (N==7 && X1->Text!="" && X2->Text!="" && X3->Text!="" &&
X4->Text!="" && X5->Text!="" && X6->Text!="" &&
X7->Text!="" && Y1->Text!="" && Y2->Text!="" &&
Y3->Text!="" && Y4->Text!="" && Y5->Text!="" &&
Y6->Text!="" && X7->Text!="")
{
X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
X55=X5->Text.ToDouble(); X66=X6->Text.ToDouble();
X77=X7->Text.ToDouble(); Y11=Y1->Text.ToDouble();
Y22=Y2->Text.ToDouble(); Y33=Y3->Text.ToDouble();
Y44=Y4->Text.ToDouble(); Y55=Y5->Text.ToDouble();
Y66=Y6->Text.ToDouble(); Y77=Y7->Text.ToDouble();

if (Edit5->Text=="")
{
MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
MB_ICONINFORMATION);

Label33->Visible=false; Label34->Visible=false; Label39->Visible=false;
Label40->Visible=false; Label37->Visible=false; Label38->Visible=false;
Label24->Visible=false; Label25->Visible=false; Label26->Visible=false;
Label27->Visible=false; Label28->Visible=false; Label29->Visible=false;
Label30->Visible=false; Label31->Visible=false; Label32->Visible=false;

Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
}

else
{
Ed1=Min(X11, Min(X22, Min(X33, Min(X44, Min(X55, Min(X66, X77))))));
Ed2=Max(X11, Max(X22, Max(X33, Max(X44, Max(X55, Max(X66, X77))))));
Ed3=Min(Y11, Min(Y22, Min(Y33, Min(Y44, Min(Y55, Min(Y66, Y77))))));
Ed4=Max(Y11, Max(Y22, Max(Y33, Max(Y44, Max(Y55, Max(Y66, Y77))))));

Image1->Visible=true;
Shape1->Visible=true;

Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
}
}

```

```

Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

Label132->Caption=Max(fabs(Ed1), fabs(Ed2));
Label124->Caption=(Label132->Caption)*(-1.0);
Label130->Caption=Label132->Caption.ToDouble()/4.0;
Label131->Caption=Label132->Caption.ToDouble()/2.0;
Label128->Caption=(Label132->Caption.ToDouble()/4.0)*3.0;
Label125->Caption=(Label128->Caption)*(-1.0);
Label126->Caption=(Label131->Caption)*(-1.0);
Label127->Caption=(Label130->Caption)*(-1.0);

Label133->Caption=Max(fabs(Ed3), fabs(Ed4));
Label138->Caption=(Label133->Caption)*(-1.0);
Label134->Caption=(Label133->Caption.ToDouble()/3.0)*2.0;
Label137->Caption=(Label133->Caption.ToDouble()/3.0);
Label139->Caption=(Label134->Caption.ToDouble()*(-1.0));
Label140->Caption=(Label137->Caption.ToDouble()*(-1.0));

if (Ed1<Ed2)//X1->Text<X2->Text)
{
    if (Edit5->Text=="")
        MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);
    else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

        x1gr=Ed1;    x2gr=Ed2;    y1gr=Ed3;    y2gr=Ed4;

        xstep=(x2gr-x1gr)/100.0;

        XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
        YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
    }//else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

else if (Ed1>Ed2)//X1->Text>X2->Text)
{

```



```

if (Edit5->Text=="")
    MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Ed2;    x2gr=Ed1;    y1gr=Ed4;    y2gr=Ed3;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
    ygr=func(xgr);
    xg=xgr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->LineTo(xg,yg);
}
} //megalos else
Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point4->Top=Image1->Top + (Image1->Height)/2 +
((Y4->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point5->Top=Image1->Top + (Image1->Height)/2 +
((Y5->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point5->Left=Image1->Left + (Image1->Width)/2 +
((X5->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point6->Top=Image1->Top + (Image1->Height)/2 +
((Y6->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point6->Left=Image1->Left + (Image1->Width)/2 +
((X6->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

```

```

    Point7->Top=Imagel->Top + (Imagel->Height)/2 +
((Y7->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
    Point7->Left=Imagel->Left + (Imagel->Width)/2 +
    ((X7->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

    Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
    Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
    Point7->Visible=true; Point8->Visible=false; Point9->Visible=false;
    Point10->Visible=false;

} //N==7

if (N==8 && X1->Text!="" && X2->Text!="" && X3->Text!="" &&
    X4->Text!="" && X5->Text!="" && X6->Text!="" &&
    X7->Text!="" && X8->Text!="" && Y1->Text!="" &&
    Y2->Text!="" && Y3->Text!="" && Y4->Text!="" &&
    Y5->Text!="" && Y6->Text!="" && X7->Text!="" && Y8->Text!="")
{
    X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
    X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
    X55=X5->Text.ToDouble(); X66=X6->Text.ToDouble();
    X77=X7->Text.ToDouble(); X88=X8->Text.ToDouble();
    Y11=Y1->Text.ToDouble(); Y22=Y2->Text.ToDouble();
    Y33=Y3->Text.ToDouble(); Y44=Y4->Text.ToDouble();
    Y55=Y5->Text.ToDouble(); Y66=Y6->Text.ToDouble();
    Y77=Y7->Text.ToDouble(); Y88=Y8->Text.ToDouble();

    if (Edit5->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);

        Label133->Visible=false; Label134->Visible=false; Label139->Visible=false;
        Label140->Visible=false; Label137->Visible=false; Label138->Visible=false;
        Label124->Visible=false; Label125->Visible=false; Label126->Visible=false;
        Label127->Visible=false; Label128->Visible=false; Label129->Visible=false;
        Label130->Visible=false; Label131->Visible=false; Label132->Visible=false;

        Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
        Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
    }

    else
    {
        Ed1=Min (X11, Min (X22, Min (X33, Min (X44, Min (X55, Min
            (X66, Min (X77, X88))))));

        Ed2=Max (X11, Max (X22, Max (X33, Max (X44, Max (X55, Max
            (X66, Max (X77, X88))))));

        Ed3=Min (Y11, Min (Y22, Min (Y33, Min (Y44, Min (Y55, Min
            (Y66, Min (Y77, Y88))))));

        Ed4=Max (Y11, Max (Y22, Max (Y33, Max (Y44, Max (Y55, Max
            (Y66, Max (Y77, Y88))))));

        Imagel->Visible=true;
        Shape1->Visible=true;

        Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
        Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
        Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
        Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
        Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

        Label32->Caption=Max (fabs (Ed1), fabs (Ed2));
        Label24->Caption=(Label32->Caption)*(-1.0);
    }
}

```

```

Label30->Caption=Label32->Caption.ToDouble()/4.0;
Label31->Caption=Label32->Caption.ToDouble()/2.0;
Label28->Caption=(Label32->Caption.ToDouble()/4.0)*3.0;
Label25->Caption=(Label28->Caption)*(-1.0);
Label26->Caption=(Label31->Caption)*(-1.0);
Label27->Caption=(Label30->Caption)*(-1.0);

Label33->Caption=Max(fabs(Ed3), fabs(Ed4));
Label38->Caption=(Label33->Caption)*(-1.0);
Label34->Caption=(Label33->Caption.ToDouble()/3.0)*2.0;
Label37->Caption=(Label33->Caption.ToDouble())/3.0;
Label39->Caption=(Label34->Caption.ToDouble()*(-1.0));
Label40->Caption=(Label37->Caption.ToDouble()*(-1.0));

if (Ed1<Ed2)//X1->Text<X2->Text)
{
if (Edit5->Text=="")
    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
        MB_ICONINFORMATION);
else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;
        Y0=Height/2;

        Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
        Image1->Canvas->Brush->Style=bsSolid;
        Image1->Canvas->FillRect(Rect(0,0,Width,Height));

        Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
        Image1->Canvas->Pen->Width=2;
        Image1->Canvas->MoveTo(X0,0);
        Image1->Canvas->LineTo(X0,Height);
        Image1->Canvas->MoveTo(0,Y0);
        Image1->Canvas->LineTo(Width,Y0);
        Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed1;    x2gr=Ed2;    y1gr=Ed3;    y2gr=Ed4;

        xstep=(x2gr-x1gr)/100.0;

        XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
        YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
    }//else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

    for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg,yg);
    }
}

else if (Ed1>Ed2)//X1->Text>X2->Text)
{
if (Edit5->Text=="")

    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
        MB_ICONINFORMATION);
else
    {
        xgr=Edit5->Text.ToDouble();

        Height=Image1->Picture->Height;
        Width=Image1->Picture->Width;

        X0=Width/2;

```

```

Y0=Height/2;

Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect(Rect(0,0,Width,Height));

Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;
x1gr=Ed2;    x2gr=Ed1;    y1gr=Ed4;    y2gr=Ed3;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg,yg);

for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
    ygr=func(xgr);
    xg=xgr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->LineTo(xg,yg);
}
} //megal0 else
Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble())*(-41))* (3/(Label133->Caption.ToDouble()))-1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point4->Top=Image1->Top + (Image1->Height)/2 +
((Y4->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point5->Top=Image1->Top + (Image1->Height)/2 +
((Y5->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point5->Left=Image1->Left + (Image1->Width)/2 +
((X5->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point6->Top=Image1->Top + (Image1->Height)/2 +
((Y6->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point6->Left=Image1->Left + (Image1->Width)/2 +
((X6->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point7->Top=Image1->Top + (Image1->Height)/2 +
((Y7->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point7->Left=Image1->Left + (Image1->Width)/2 +
((X7->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point8->Top=Image1->Top + (Image1->Height)/2 +
((Y8->Text.ToDouble())*(-41))* (2.9/(Label133->Caption.ToDouble()))-1;
Point8->Left=Image1->Left + (Image1->Width)/2 +
((X8->Text.ToDouble())*46)* (3.9/(Label132->Caption.ToDouble()))-1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;

```

```

Point7->Visible=true;Point8->Visible=true; Point9->Visible=false;
Point10->Visible=false;

} //N==8

if (N==9 && X1->Text!="" && X2->Text!="" && X3->Text!="" &&
    X4->Text!="" && X5->Text!="" && X6->Text!="" &&
    X7->Text!="" && X8->Text!="" && X9->Text!="" &&
    Y1->Text!="" && Y2->Text!="" && Y3->Text!="" &&
    Y4->Text!="" && Y5->Text!="" && Y6->Text!="" &&
    X7->Text!="" && X8->Text!="" && Y9->Text!="" )
{
    X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
    X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
    X55=X5->Text.ToDouble(); X66=X6->Text.ToDouble();
    X77=X7->Text.ToDouble(); X88=X8->Text.ToDouble();
    X99=X9->Text.ToDouble(); Y11=Y1->Text.ToDouble();
    Y22=Y2->Text.ToDouble(); Y33=Y3->Text.ToDouble();
    Y44=Y4->Text.ToDouble(); Y55=Y5->Text.ToDouble();
    Y66=Y6->Text.ToDouble(); Y77=Y7->Text.ToDouble();
    Y88=Y8->Text.ToDouble(); Y99=Y9->Text.ToDouble();

    if (Edit5->Text=="")
    {
        MessageBox(NULL,"Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
            MB_ICONINFORMATION);

        Label133->Visible=false;Label134->Visible=false;Label139->Visible=false;
        Label140->Visible=false;Label137->Visible=false;Label138->Visible=false;
        Label124->Visible=false;Label125->Visible=false;Label126->Visible=false;
        Label127->Visible=false;Label128->Visible=false;Label129->Visible=false;
        Label130->Visible=false;Label131->Visible=false;Label132->Visible=false;

        Shape2->Visible=false;Shape3->Visible=false; Shape18->Visible=false;
        Shape6->Visible=false;Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false;Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false;Shape13->Visible=false;Shape14->Visible=false;
        Shape15->Visible=false;Shape16->Visible=false;Shape17->Visible=false;
    }
    else
    {
        Ed1=Min(X11,Min(X22,Min(X33,Min(X44,Min(X55,Min
            (X66,Min(X77,Min(X88,X99))))))););

        Ed2=Max(X11,Max(X22,Max(X33,Max(X44,Max(X55,Max
            (X66,Max(X77,Max(X88,X99))))))););

        Ed3=Min(Y11,Min(Y22,Min(Y33,Min(Y44,Min(Y55,Min
            (Y66,Min(Y77,Min(Y88,Y99))))))););

        Ed4=Max(Y11,Max(Y22,Max(Y33,Max(Y44,Max(Y55,Max
            (Y66,Max(Y77,Max(Y88,Y99))))))););

        Image1->Visible=true;
        Shape1->Visible=true;

        Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        Label133->Visible=true;Label134->Visible=true; Label139->Visible=true;
        Label140->Visible=true;Label137->Visible=true; Label138->Visible=true;
        Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
        Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
        Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

        Label132->Caption=Max(fabs(Ed1),fabs(Ed2));
        Label124->Caption=(Label132->Caption)*(-1.0);
        Label130->Caption=Label132->Caption.ToDouble()/4.0;
        Label131->Caption=Label132->Caption.ToDouble()/2.0;
        Label128->Caption=(Label132->Caption.ToDouble()/4.0)*3.0;
        Label125->Caption=(Label128->Caption)*(-1.0);
        Label126->Caption=(Label131->Caption)*(-1.0);
        Label127->Caption=(Label130->Caption)*(-1.0);
    }
}

```

```

Label33->Caption=Max(fabs(Ed3), fabs(Ed4));
Label38->Caption=(Label33->Caption)*(-1.0);
Label34->Caption=(Label33->Caption.ToDouble()/3.0)*2.0;
Label37->Caption=(Label33->Caption.ToDouble())/3.0;
Label39->Caption=(Label34->Caption.ToDouble()*(-1.0);
Label40->Caption=(Label37->Caption.ToDouble()*(-1.0);

if(Ed1<Ed2)
{
if(Edit5->Text=="")
    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed1;    x2gr=Ed2;    y1gr=Ed3;    y2gr=Ed4;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
} //else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg, yg);

    for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg, yg);
    }
}

else if(Ed1>Ed2)
{
if(Edit5->Text=="")
    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK|
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
    Image1->Canvas->Pen->Width=2;

```

```

Image1->Canvas->MoveTo(X0,0);
Image1->Canvas->LineTo(X0,Height);
Image1->Canvas->MoveTo(0,Y0);
Image1->Canvas->LineTo(Width,Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed2;    x2gr=Ed1;    y1gr=Ed4;    y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max(fabs(x1gr),fabs(x2gr));
YScale=Height/2/Max(fabs(y1gr),fabs(y2gr));
} //else

ygr=func(x1gr);
xg=x1gr*XScale+X0;
yg=-ygr*YScale+Y0;
Image1->Canvas->MoveTo(xg,yg);

for(xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
    ygr=func(xgr);
    xg=xgr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->LineTo(xg,yg);
}
} //megalo else

Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble())*(-41))*(3/(Label133->Caption.ToDouble()))-1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble())*(-41))*(3/(Label133->Caption.ToDouble()))-1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point4->Top=Image1->Top + (Image1->Height)/2 +
((Y4->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point5->Top=Image1->Top + (Image1->Height)/2 +
((Y5->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point5->Left=Image1->Left + (Image1->Width)/2 +
((X5->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point6->Top=Image1->Top + (Image1->Height)/2 +
((Y6->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point6->Left=Image1->Left + (Image1->Width)/2 +
((X6->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point7->Top=Image1->Top + (Image1->Height)/2 +
((Y7->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point7->Left=Image1->Left + (Image1->Width)/2 +
((X7->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point8->Top=Image1->Top + (Image1->Height)/2 +
((Y8->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point8->Left=Image1->Left + (Image1->Width)/2 +
((X8->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point9->Top=Image1->Top + (Image1->Height)/2 +
((Y9->Text.ToDouble())*(-41))*(2.9/(Label133->Caption.ToDouble()))-1;
Point9->Left=Image1->Left + (Image1->Width)/2 +
((X9->Text.ToDouble())*46)*(3.9/(Label132->Caption.ToDouble()))-1;

Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=true; Point8->Visible=true; Point9->Visible=true;

```

```

Point10->Visible=false;

} //N==9

if (N==10 && X1->Text!="" && X2->Text!="" && X3->Text!="" &&
    X4->Text!="" && X5->Text!="" && X6->Text!="" &&
    X7->Text!="" && X8->Text!="" && X9->Text!="" &&
    X10->Text!="" && Y1->Text!="" && Y2->Text!="" &&
    Y3->Text!="" && Y4->Text!="" && Y5->Text!="" &&
    Y6->Text!="" && Y7->Text!="" && Y8->Text!="" &&
    Y9->Text!="" && Y10->Text!="")
{
    X11=X1->Text.ToDouble(); X22=X2->Text.ToDouble();
    X33=X3->Text.ToDouble(); X44=X4->Text.ToDouble();
    X55=X5->Text.ToDouble(); X66=X6->Text.ToDouble();
    X77=X7->Text.ToDouble(); X88=X8->Text.ToDouble();
    X99=X9->Text.ToDouble(); X100=X10->Text.ToDouble();
    Y11=Y1->Text.ToDouble(); Y22=Y2->Text.ToDouble();
    Y33=Y3->Text.ToDouble(); Y44=Y4->Text.ToDouble();
    Y55=Y5->Text.ToDouble(); Y66=Y6->Text.ToDouble();
    Y77=Y7->Text.ToDouble(); Y88=Y8->Text.ToDouble();
    Y99=Y9->Text.ToDouble(); Y100=Y10->Text.ToDouble();

    if (Edit5->Text=="")
    {
        MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
            MB_ICONINFORMATION);

        Label133->Visible=false; Label134->Visible=false; Label139->Visible=false;
        Label140->Visible=false; Label137->Visible=false; Label138->Visible=false;
        Label124->Visible=false; Label125->Visible=false; Label126->Visible=false;
        Label127->Visible=false; Label128->Visible=false; Label129->Visible=false;
        Label130->Visible=false; Label131->Visible=false; Label132->Visible=false;

        Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
        Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;
    }

    else
    {
        Ed1=Min(X11, Min(X22, Min(X33, Min(X44, Min(X55, Min
            (X66, Min(X77, Min(X88, Min(X99, X100)))))))));

        Ed2=Max(X11, Max(X22, Max(X33, Max(X44, Max(X55, Max
            (X66, Max(X77, Max(X88, Max(X99, X100)))))))));

        Ed3=Min(Y11, Min(Y22, Min(Y33, Min(Y44, Min(Y55, Min
            (Y66, Min(Y77, Min(Y88, Min(Y99, Y100)))))))));

        Ed4=Max(Y11, Max(Y22, Max(Y33, Max(Y44, Max(Y55, Max
            (Y66, Max(Y77, Max(Y88, Max(Y99, Y100)))))))));

        Image1->Visible=true;
        Shape1->Visible=true;

        Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        Label133->Visible=true; Label134->Visible=true; Label139->Visible=true;
        Label140->Visible=true; Label137->Visible=true; Label138->Visible=true;
        Label124->Visible=true; Label125->Visible=true; Label126->Visible=true;
        Label127->Visible=true; Label128->Visible=true; Label129->Visible=true;
        Label130->Visible=true; Label131->Visible=true; Label132->Visible=true;

        Label32->Caption=Max(fabs(Ed1), fabs(Ed2));
        Label24->Caption=(Label32->Caption)*(-1.0);
        Label30->Caption=Label32->Caption.ToDouble()/4.0;
        Label31->Caption=Label32->Caption.ToDouble()/2.0;
        Label28->Caption=(Label32->Caption.ToDouble()/4.0)*3.0;
        Label25->Caption=(Label28->Caption)*(-1.0);
    }
}

```



```

Label26->Caption=(Label31->Caption)*(-1.0);
Label27->Caption=(Label30->Caption)*(-1.0);

Label33->Caption=Max(fabs(Ed3), fabs(Ed4));
Label38->Caption=(Label33->Caption)*(-1.0);
Label34->Caption=(Label33->Caption.ToDouble()/3.0)*2.0;
Label37->Caption=(Label33->Caption.ToDouble()/3.0);
Label39->Caption=(Label34->Caption.ToDouble()*(-1.0));
Label40->Caption=(Label37->Caption.ToDouble()*(-1.0));

if (Ed1<Ed2) //X1->Text<X2->Text)
{
if (Edit5->Text=="")

    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
    Image1->Canvas->Pen->Width=2;
    Image1->Canvas->MoveTo(X0,0);
    Image1->Canvas->LineTo(X0,Height);
    Image1->Canvas->MoveTo(0,Y0);
    Image1->Canvas->LineTo(Width,Y0);
    Image1->Canvas->Pen->Color=LineColor;

    x1gr=Ed1;    x2gr=Ed2;    y1gr=Ed3;    y2gr=Ed4;

    xstep=(x2gr-x1gr)/100.0;

    XScale=Width/2/Max(fabs(x1gr), fabs(x2gr));
    YScale=Height/2/Max(fabs(y1gr), fabs(y2gr));
} //else

    ygr=func(x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo(xg, yg);

    for (xgr=x1gr; xgr<=x2gr; xgr+=xstep)
    {
        ygr=func(xgr);
        xg=xgr*XScale+X0;
        yg=-ygr*YScale+Y0;
        Image1->Canvas->LineTo(xg, yg);
    }
}

else if (Ed1>Ed2) //X1->Text>X2->Text)
{
if (Edit5->Text=="")

    MessageBox(NULL, "Δεν υπάρχουν δεδομένα", "Προσοχή", MB_OK |
                MB_ICONINFORMATION);
else
{
    xgr=Edit5->Text.ToDouble();

    Height=Image1->Picture->Height;
    Width=Image1->Picture->Width;

    X0=Width/2;
    Y0=Height/2;

    Image1->Canvas->Brush->Color=ImageCanBrush;//clNavy;

```

```

Image1->Canvas->Brush->Style=bsSolid;
Image1->Canvas->FillRect (Rect (0,0,Width,Height));

Image1->Canvas->Pen->Color=ImageCanPen;//clMaroon;
Image1->Canvas->Pen->Width=2;
Image1->Canvas->MoveTo (X0,0);
Image1->Canvas->LineTo (X0,Height);
Image1->Canvas->MoveTo (0,Y0);
Image1->Canvas->LineTo (Width,Y0);
Image1->Canvas->Pen->Color=LineColor;

x1gr=Ed2;    x2gr=Ed1;    y1gr=Ed4;    y2gr=Ed3;

xstep=(x2gr-x1gr)/100.0;

XScale=Width/2/Max (fabs (x1gr) , fabs (x2gr) );
YScale=Height/2/Max (fabs (y1gr) , fabs (y2gr) );
} //else

    ygr=func (x1gr);
    xg=x1gr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->MoveTo (xg,yg);

for (xgr=x1gr;xgr<=x2gr;xgr+=xstep)
{
    ygr=func (xgr);
    xg=xgr*XScale+X0;
    yg=-ygr*YScale+Y0;
    Image1->Canvas->LineTo (xg,yg);
}
} //megalo else
Point1->Top=Image1->Top + (Image1->Height)/2 +
((Y1->Text.ToDouble ()) * (-41)) * (3 / (Label133->Caption.ToDouble ())) -1;
Point1->Left=Image1->Left + (Image1->Width)/2 +
((X1->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point2->Top=Image1->Top + (Image1->Height)/2 +
((Y2->Text.ToDouble ()) * (-41)) * (3 / (Label133->Caption.ToDouble ())) -1;
Point2->Left=Image1->Left + (Image1->Width)/2 +
((X2->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point3->Top=Image1->Top + (Image1->Height)/2 +
((Y3->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point3->Left=Image1->Left + (Image1->Width)/2 +
((X3->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point4->Top=Image1->Top + (Image1->Height)/2 +
((Y4->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point4->Left=Image1->Left + (Image1->Width)/2 +
((X4->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point5->Top=Image1->Top + (Image1->Height)/2 +
((Y5->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point5->Left=Image1->Left + (Image1->Width)/2 +
((X5->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point6->Top=Image1->Top + (Image1->Height)/2 +
((Y6->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point6->Left=Image1->Left + (Image1->Width)/2 +
((X6->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point7->Top=Image1->Top + (Image1->Height)/2 +
((Y7->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point7->Left=Image1->Left + (Image1->Width)/2 +
((X7->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point8->Top=Image1->Top + (Image1->Height)/2 +
((Y8->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point8->Left=Image1->Left + (Image1->Width)/2 +
((X8->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

Point9->Top=Image1->Top + (Image1->Height)/2 +
((Y9->Text.ToDouble ()) * (-41)) * (2.9 / (Label133->Caption.ToDouble ())) -1;
Point9->Left=Image1->Left + (Image1->Width)/2 +
((X9->Text.ToDouble ()) * 46) * (3.9 / (Label132->Caption.ToDouble ())) -1;

```

```

    Point10->Top=Image1->Top + (Image1->Height)/2 +
((Y10->Text.ToDouble())*(-41))*(2.9/(Label33->Caption.ToDouble()))-1;
    Point10->Left=Image1->Left + (Image1->Width)/2 +
((X10->Text.ToDouble())*46)*(3.9/(Label32->Caption.ToDouble()))-1;

    Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
    Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
    Point7->Visible=true; Point8->Visible=true; Point9->Visible=true;
    Point10->Visible=true;

} //N==10

Label24->Width=17;
Label25->Width=17;
Label26->Width=17;
Label27->Width=20;
Label28->Width=20;
Label29->Width=20;
Label30->Width=20;
Label31->Width=20;
Label32->Width=20;
Label33->Width=18;
Label34->Width=17;
Label37->Width=17;
Label38->Width=19;
Label39->Width=20;
Label40->Width=20;
}
//-----
void __fastcall TForm16::N3Click(TObject *Sender)
{
    if (N3->Checked)
    {
        Label33->Visible=false; Label34->Visible=false; Label39->Visible=false;
        Label40->Visible=false; Label37->Visible=false; Label38->Visible=false;
        Label24->Visible=false; Label25->Visible=false; Label26->Visible=false;
        Label27->Visible=false; Label28->Visible=false; Label29->Visible=false;
        Label30->Visible=false; Label31->Visible=false; Label32->Visible=false;

        Shape2->Visible=false; Shape3->Visible=false; Shape18->Visible=false;
        Shape6->Visible=false; Shape7->Visible=false; Shape8->Visible=false;
        Shape9->Visible=false; Shape10->Visible=false; Shape11->Visible=false;
        Shape12->Visible=false; Shape13->Visible=false; Shape14->Visible=false;
        Shape15->Visible=false; Shape16->Visible=false; Shape17->Visible=false;

        N3->Checked=false;
        Image1->Visible=false;
        Shape1->Visible=false;

        Point1->Visible=false; Point2->Visible=false; Point3->Visible=false;
        Point4->Visible=false; Point5->Visible=false; Point6->Visible=false;
        Point7->Visible=false; Point8->Visible=false; Point9->Visible=false;
        Point10->Visible=false;
    }

    else
    {
        Label33->Visible=true; Label34->Visible=true; Label39->Visible=true;
        Label40->Visible=true; Label37->Visible=true; Label38->Visible=true;
        Label24->Visible=true; Label25->Visible=true; Label26->Visible=true;
        Label27->Visible=true; Label28->Visible=true; Label29->Visible=true;
        Label30->Visible=true; Label31->Visible=true; Label32->Visible=true;

        Shape2->Visible=true; Shape3->Visible=true; Shape18->Visible=true;
        Shape6->Visible=true; Shape7->Visible=true; Shape8->Visible=true;
        Shape9->Visible=true; Shape10->Visible=true; Shape11->Visible=true;
        Shape12->Visible=true; Shape13->Visible=true; Shape14->Visible=true;
        Shape15->Visible=true; Shape16->Visible=true; Shape17->Visible=true;

        N3->Checked=true;
        Image1->Visible=true;
        Shape1->Visible=true;

        if (X1->Text="" || X2->Text="" || X3->Text="" || X4->Text="" ||
            X5->Text="" || X6->Text="" || X7->Text="" || X8->Text="" ||
            X9->Text="" || X10->Text="" || Y1->Text="" || Y2->Text="" ||

```

```

Y3->Text=="" || Y4->Text=="" || X5->Text=="" || X6->Text=="" ||
X7->Text=="" || Y8->Text=="" || Y9->Text=="" || Y10->Text=="")

MessageBox(NULL,"Δεν Υπάρχουν Στοιχεία", "Προσοχή", MB_ICONINFORMATION);

else if (Edit1->Text.ToDouble()==2)
{Point1->Visible=true; Point2->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==3)
{Point1->Visible=true; Point2->Visible=true;
Point3->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==4)
{Point1->Visible=true; Point2->Visible=true;
Point3->Visible=true; Point4->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==5)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==6)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Button1->Click();}
else if (Edit1->Text.ToDouble()==7)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==8)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=true; Point8->Visible=true; Button1->Click();}
else if (Edit1->Text.ToDouble()==9)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=true; Point8->Visible=true; Point9->Visible=true;
Button1->Click();}
else if (Edit1->Text.ToDouble()==10)
{Point1->Visible=true; Point2->Visible=true; Point3->Visible=true;
Point4->Visible=true; Point5->Visible=true; Point6->Visible=true;
Point7->Visible=true; Point8->Visible=true; Point9->Visible=true;
Point10->Visible=true; Button1->Click();}
}
}
//-----
void __fastcall TForm16::N13Click(TObject *Sender)
{
if (ColorDialog1->Execute())
LineColor = ColorDialog1->Color;
Button1->Click();
}
//-----
void __fastcall TForm16::N14Click(TObject *Sender)
{
if (ColorDialog1->Execute())
ImageCanPen = ColorDialog1->Color;
Button1->Click();
}
//-----
void __fastcall TForm16::N15Click(TObject *Sender)
{
if (ColorDialog1->Execute())
ImageCanBrush = ColorDialog1->Color;
Button1->Click();
}
//-----
void __fastcall TForm16::N16Click(TObject *Sender)
{
if (ColorDialog1->Execute())
{
Shape2->Brush->Color = ColorDialog1->Color;
Shape3->Brush->Color = ColorDialog1->Color;
Shape18->Brush->Color = ColorDialog1->Color;
Shape6->Brush->Color = ColorDialog1->Color;
Shape7->Brush->Color = ColorDialog1->Color;
Shape8->Brush->Color = ColorDialog1->Color;
Shape9->Brush->Color = ColorDialog1->Color;
Shape10->Brush->Color = ColorDialog1->Color;
Shape11->Brush->Color = ColorDialog1->Color;
Shape12->Brush->Color = ColorDialog1->Color;
Shape13->Brush->Color = ColorDialog1->Color;
}
}
}

```

```

Shape14->Brush->Color = ColorDialog1->Color;
Shape15->Brush->Color = ColorDialog1->Color;
Shape16->Brush->Color = ColorDialog1->Color;
Shape17->Brush->Color = ColorDialog1->Color;
Label133->Font->Color = ColorDialog1->Color;
Label134->Font->Color = ColorDialog1->Color;
Label137->Font->Color = ColorDialog1->Color;
Label138->Font->Color = ColorDialog1->Color;
Label139->Font->Color = ColorDialog1->Color;
Label140->Font->Color = ColorDialog1->Color;
Label124->Font->Color = ColorDialog1->Color;
Label125->Font->Color = ColorDialog1->Color;
Label126->Font->Color = ColorDialog1->Color;
Label127->Font->Color = ColorDialog1->Color;
Label128->Font->Color = ColorDialog1->Color;
Label129->Font->Color = ColorDialog1->Color;
Label130->Font->Color = ColorDialog1->Color;
Label131->Font->Color = ColorDialog1->Color;
Label132->Font->Color = ColorDialog1->Color;

Label124->Width=25;
Label125->Width=25;
Label126->Width=25;
Label127->Width=25;
Label128->Width=17;
Label129->Width=18;
Label130->Width=17;
Label131->Width=17;
Label132->Width=17;
Label133->Width=17;
Label134->Width=17;
Label137->Width=17;
Label138->Width=20;
Label139->Width=20;
Label140->Width=20;
}
}
//-----
void __fastcall TForm16::FormCreate(TObject *Sender)
{
    int Height=Image1->Picture->Height;
    int Width=Image1->Picture->Width;

    Image1->Canvas->Brush->Color=ImageCanBrush;
    Image1->Canvas->Brush->Style=bsSolid;
    Image1->Canvas->FillRect(Rect(0,0,Width,Height));

    Label133->Caption=="--";Label134->Caption=="--";Label137->Caption=="--";
    Label138->Caption=="--";Label139->Caption=="--";Label140->Caption=="--";
    Label124->Caption=="--";Label125->Caption=="--";Label126->Caption=="--";
    Label127->Caption=="--";Label128->Caption=="--";Label129->Caption=="--";
    Label130->Caption=="--";Label131->Caption=="--";Label132->Caption=="--";
}
//-----
void __fastcall TForm16::N9Click(TObject *Sender)
{
    BitBtn2->Click();
}

```

# Βιβλιογραφία

## Έντυπη μορφή

William H.Press, Saul A.Teukolsky, William T.Vatterling, Brian P.Flannery, **Αριθμητικές μέθοδοι στην γλώσσα C++**, 2<sup>η</sup> έκδοση, Εκδόσεις Cambridge U.P., 2002.

Γεώργιος Δ. Ακρίβης, Βασίλειος Α. Δουγαλής, **Εισαγωγή στην αριθμητική ανάλυση**, 4<sup>η</sup> έκδοση, Πανεπιστημιακές Εκδόσεις Κρήτης, 2002.

Πάρις Μαστοροκόστας, **Αριθμητικές Μέθοδοι σε Επιστημονικές Εφαρμογές**, ΤΕΙ Σερρών - Τμήμα πληροφορικής & Επικοινωνιών, Σέρρες 2001

Σπύρος Καζαρλής, **Οπτικός Προγραμματισμός**, ΤΕΙ Σερρών - Τμήμα Πληροφορικής & Επικοινωνιών, Σέρρες 2004

Σπύρος Καζαρλής, **Οπτικός Προγραμματισμός (Σημειώσεις Εργαστηρίου)**, ΤΕΙ Σερρών - Τμήμα Πληροφορικής & Επικοινωνιών, Σέρρες 2003

Stanley B. Lippman, **Essential C++**, September 12, 2002, Addison Wesley

Stanley B. Lippman, Josee Lajoie, **C++ Primer**, Third Edition, Addison Wesley

Jan Walter, Danny Kalev, Michael J. Tobler, Paul Snaith, Andrei Kossorokov, Scott Roberts, **C++ How To**, The Waite Group's

Bob Swart, Mark Cashman, Paul Gustavson and Jarrod Hollingworth, **Borland C++ Builder 6 Developer's Guide**, 2003, Michael Stephens

Kate Gregory, **Special Edition Using Visual C++6**, August 1998, QUE

## Ηλεκτρονική μορφή

[www.nr.com](http://www.nr.com) : Ιστοσελίδα αριθμητικών μεθόδων του Πανεπιστημίου Oxford.