

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ  
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ  
ΥΠΟΛΟΓΙΣΤΩΝ

## ΕΡΓΑΣΙΑ 3 ΔΟΜΕΣ ΕΛΕΓΧΟΥ

### ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ :

---

**ΟΝΟΜΑΤΕΠΩΝΥΜΟ :** ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ

**ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ :** 19390005

**ΚΑΤΑΣΤΑΣΗ ΦΟΙΤΗΤΗ :** ΠΡΟΠΤΥΧΙΑΚΟ

**ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ :** ΠΑΔΑ

**ΤΜΗΜΑ ΕΡΓΑΣΤΗΡΙΟΥ :** M2

**ΚΑΘΗΓΗΤΗΣ ΕΡΓΑΣΤΗΡΙΟΥ :** ΓΕΩΡΓΙΟΣ ΜΕΛΕΤΙΟΥ

**ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ ΤΗΣ ΕΡΓΑΣΙΑΣ :** 23/11/2021

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΠΕΡΙΕΧΟΜΕΝΑ

### ΣΕΛΙΔΕΣ 3-5:

#### ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ:

ΘΕΜΑ 1a) Τι γνωρίζετε για την εντολή «if»; (ΣΕΛΙΔΑ 3)

ΘΕΜΑ 1b) Τι γνωρίζετε για την εντολή «switch»; (ΣΕΛΙΔΕΣ 4-5)

### ΣΕΛΙΔΕΣ 5-6:

#### ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΠΗΓΑΙΟΥΣ ΚΩΔΙΚΕΣ

|                      |               |
|----------------------|---------------|
| ΘΕΜΑ 2               | (ΣΕΛΙΔΕΣ 5-6) |
| Επισήμανση           | (ΣΕΛΙΔΑ 5)    |
| Πρόγραμμα «if.c»     | (ΣΕΛΙΔΑ 5)    |
| Πρόγραμμα «if1.c»    | (ΣΕΛΙΔΕΣ 5-6) |
| Πρόγραμμα «switch.c» | (ΣΕΛΙΔΕΣ 6)   |

### ΣΕΛΙΔΕΣ 6-26:

#### ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ

|                 |                 |
|-----------------|-----------------|
| ΘΕΜΑ 3          | (ΣΕΛΙΔΕΣ 6-17)  |
| Επισήμανση «C1» | (ΣΕΛΙΔΕΣ 6)     |
| Πρόγραμμα «C1»  | (ΣΕΛΙΔΕΣ 6-8)   |
| Τεκμηρίωση «C1» | (ΣΕΛΙΔΕΣ 9-17)  |
| {               |                 |
| Ζητούμενο       | (ΣΕΛΙΔΑ 9)      |
| Δομή            | (ΣΕΛΙΔΑ 9)      |
| Μεταβλητές      | (ΣΕΛΙΔΑ 10)     |
| Διάσχιση        | (ΣΕΛΙΔΕΣ 10-13) |
| Παραδείγματα    | (ΣΕΛΙΔΕΣ 13-16) |
| Παρατηρήσεις    | (ΣΕΛΙΔΕΣ 16-17) |
| }               |                 |
| ΘΕΜΑ 4          | (ΣΕΛΙΔΕΣ 17-26) |
| Επισήμανση «C2» | (ΣΕΛΙΔΑ 17)     |
| Πρόγραμμα «C2»  | (ΣΕΛΙΔΕΣ 17-18) |
| Τεκμηρίωση «C2» | (ΣΕΛΙΔΕΣ 18-26) |
| {               |                 |
| Ζητούμενο       | (ΣΕΛΙΔΑ 18)     |
| Δομή            | (ΣΕΛΙΔΑ 19)     |
| Μεταβλητές      | (ΣΕΛΙΔΑ 19)     |

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Διάσχιση  
Παραδείγματα  
Παρατηρήσεις  
}

(ΣΕΛΙΔΕΣ 19-21)

(ΣΕΛΙΔΕΣ 21-25)

(ΣΕΛΙΔΕΣ 25-26)

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ

### Θέμα 1

#### a) Τι γνωρίζετε για την εντολή «if»:

Η εντολή «if» είναι μία εντολή που ελέγχει το αποτέλεσμα μίας παράστασης, ώστε να εκτελέσει τις κατάλληλες εντολές, ανάλογα με αυτό το αποτέλεσμα. Για μία παράσταση «p1» η δομή της «if» είναι η εξής :

```
if (p1) printf ("Hello world");  
printf ("End of if");
```

Πιο αναλυτικά, αν η παράσταση «p1» παράγει αποτέλεσμα διάφορο του μηδενός (0), τότε θα εκτελεστεί η εντολή που είναι μέσα στην «if» και στην συγκεκριμένη περίπτωση είναι η «printf ("Hello world");», όπου θα τυπωθεί το μήνυμα «Hello world». Ειδαλλώς, αν το αποτέλεσμα που παράγει είναι μηδέν (0), τότε θα εκτελεστούν οι εντολές, εφόσον και υπάρχουν, που δεν είναι μέσα στην «if» (Στην συγκεκριμένη περίπτωση θα εκτελεστεί η εντολή «printf ("End of if");», όπου θα τυπωθεί το μήνυμα «End of if»).

Η παράσταση «p1» μπορεί να περιέχει οποιαδήποτε αριθμητική, σχεσιακή, λογική, δυαδική πράξη που παράγει αποτέλεσμα «0» και διάφορο του «0». Για παράδειγμα μπορεί να είναι μία σχεσιακή συνθήκη ( $x > y$ ), μία αριθμητική πράξη ( $x + y$ ), μία ανάθεση μεταβλητής ( $x = 1$ ), μία λογική συνθήκη ( $x \&\& y$ ), μία σταθερά (5), ένας χαρακτήρας ('A'), μία συμβολοσειρά ('Hello world') κ.ο.κ.

Εκτός από την εντολή «if» υπάρχει και η εντολή «if-else» που είναι, επίσης, μία εντολή ελέγχου. Για μία παράσταση «p1» η δομή της «if-else» είναι η εξής :

```
if (p1)  
printf ("Hello world");  
else  
printf ("Goodbye world");
```

Πιο αναλυτικά, αν η παράσταση «p1» παράγει αποτέλεσμα διάφορο του μηδενός (0), τότε θα εκτελεστεί η εντολή που είναι μέσα στην «if» και στην συγκεκριμένη περίπτωση είναι η «printf ("Hello world");», όπου θα τυπωθεί το μήνυμα «Hello world». Ειδαλλώς, αν το αποτέλεσμα που παράγει είναι μηδέν (0), τότε θα εκτελεστούν οι εντολές που είναι μέσα στην «else» και στην συγκεκριμένη περίπτωση είναι η «printf ("Goodbye world");», όπου θα τυπωθεί το μήνυμα «Goodbye world»;».

Μέσα σε μία «if» μπορούμε να βάλουμε άλλη μία «if» ή και «if-else».

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Για παράδειγμα :

```
1 if (x >= 0)
2 {
3     if (x == 0)
4         printf ("zero");
5     else
6         printf ("positive");
7 }
8 else
9     printf ("negative");
```

Πιο αναλυτικά, η «if» της γραμμής 1 ελέγχει αν η παράσταση « $x \geq 0$ » παράγει αποτέλεσμα «0» ή διάφορο του «0». Αν παράγει αποτέλεσμα «0», τότε εκτελείται η εντολή μέσα στην «if» που είναι μία άλλη «if» (γραμμή 3). Η «if» της γραμμής 3 ελέγχει αν η παράσταση « $x = 0$ » παράγει αποτέλεσμα «0» ή διάφορο του «0». Αν παράγει αποτέλεσμα «0», τότε εκτελείται η εντολή μέσα στην «if» που είναι η «printf ("zero");», όπου θα τυπώσει το μήνυμα «zero» (γραμμή 4), δηλαδή, ότι το περιεχόμενο της μεταβλητής «x» είναι το «0». Ειδαλλώς, εκτελείται η εντολή μέσα στην «else» που είναι η «printf ("positive");», όπου θα τυπώσει το μήνυμα «positive», δηλαδή, ότι το περιεχόμενο της μεταβλητής «x» είναι θετικός αριθμός. Από την άλλη πλευρά, αν η αρχική συνθήκη της πρώτης «if» ( $x \geq 0$ ) της γραμμής 1 παράγει αποτέλεσμα διάφορο του «0», τότε εκτελείται η εντολή μέσα στην «else» της γραμμής 9 που είναι η εντολή «printf ("negative");», δηλαδή, ότι το περιεχόμενο της μεταβλητής «x» είναι αρνητικός αριθμός.

## **b) Τι γνωρίζετε για την εντολή «switch»:**

Η εντολή «switch» είναι μία εντολή που ελέγχει την τιμή μίας ακέραιας σταθεράς, ώστε να εκτελέσει εντολές, ανάλογα με την τιμή αυτή. Για μία ακέραια σταθερά «x» η δομή της «switch» είναι :

```
switch (x)
{
    case 1 : printf ("Hello");
    case 2 : printf ("world");
    default : printf ("\n");
}
```

Πιο αναλυτικά, η «switch» ελέγχει την τιμή της ακέραιας μεταβλητής «x». Σε περίπτωση που η μεταβλητή περιέχει την τιμή «1», τότε εκτελούνται οι εντολές στην «case 1», στην «case 2» και στην «default» που στην συγκεκριμένη περίπτωση είναι οι «printf ("Hello");», «printf ("world");» και «printf ("\n");» αντίστοιχα, όπου θα τυπωθούν τα μηνύματα «Hello», «world» και ο χαρακτήρας διαφυγής της αλλαγής γραμμής «\n». Σε περίπτωση που η μεταβλητή περιέχει την τιμή «2», τότε εκτελούνται οι εντολές στην «case 2» και στην «default». Τέλος, στην περίπτωση που η μεταβλητή περιέχει μία τιμή πέρα από τις τιμές που περιέχονται στις «case» (1 και 2), εκτελούνται μόνο οι εντολές στην

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

«default».

Επιπρόσθετα, υπάρχει η εντολή «break» που ουσιαστικά διακόπτει την ροή του προγράμματος μέσα στο παράθυρο εντολών της «switch». Για παράδειγμα:

```
switch (x)
{
    case 1 : printf ("Hello"); break;
    case 2 : printf ("world"); break;
    default : printf ("\n");
}
```

Πιο αναλυτικά, η «switch» ελέγχει την τιμή της ακέραιας μεταβλητής «x». Σε περίπτωση που η μεταβλητή περιέχει την τιμή «1», τότε εκτελούνται οι εντολές στην «case 1», δηλαδή, η «printf ("Hello");», όπου θα τυπωθεί το μήνυμα «Hello» και η «break;». Στη συνέχεια, η ροή του προγράμματος μεταφέρεται εκτός του παραθύρου εντολών της «switch» ανάμεσα από τα άγκιστρα «{ , }». Το ίδιο ισχύει και για την περίπτωση που η μεταβλητή «x» περιέχει την τιμή «2», όπου θα εκτελεστούν οι εντολές στην «case 2» και η εντολή «break;».

## ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΠΗΓΑΙΟΥΣ ΚΩΔΙΚΕΣ

### Θέμα 2

#### Επισήμανση

Σε αυτή την ενότητα καταγράφονται τα σχόλια και οι παρατηρήσεις σχετικά με τα προγράμματα «if.c», «if1.c» και «switch.c».

#### Πρόγραμμα «if.c»

Το πρόγραμμα «if.c» βασισμένο σε εντολές ελέγχου «if-else» διαβά-ζει από την «standard» είσοδο τρεις ακεραίους αριθμούς και βρίσκει τον μέγιστο αριθμό, καθώς και την σειρά που εισήχθη. Ο έλεγχος αυτός γίνεται με εμφωλευμένες εντολές «if-else», όπου σε κάθε περίπτωση ελέγχονται οι τρεις αυτοί αριθμοί που δίνει ο χρήστης για το ποιος είναι ο μεγαλύτερος και με ποια ή ποιες σειρές εισήχθη ή εισήχθησαν, καθώς ο ίδιος αριθμός μπορεί να έχει διαβαστεί πάνω από μία φορά. Το πρόγραμμα επιτυγχάνει να δείξει την λειτουργία των εμφωλευμένων εντολών «if-else».

#### Πρόγραμμα «if1.c»

Το πρόγραμμα «if1.c» βασισμένο στην εντολή ελέγχου «if» και του

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

τριαδικού τελεστή «? :» εκτελεί τον ίδιο αλγόριθμο με αυτόν του προγράμματος «if.c» (βλ. Πρόγραμμα «if.c»). Η διαφορά έγκειται στην σύνταξη και στις εντολές ελέγχου. Πιο αναλυτικά, στο πρόγραμμα «if1.c» ορίζεται μία ακέραια μεταβλητή «Max», όπου αρχική τιμή έχει την τιμή του πρώτου ακέραιου αριθμού που διαβάζεται από την «standard» είσοδο και δηλώνει τον μέγιστο αριθμό. Στη συνέχεια, με μια «if» γίνεται ένας έλεγχος μεταξύ της «Max» και του δεύτερου αριθμού, για να διαπιστωθεί αν ο αυτός είναι ο μέγιστος. Με την χρήση του τριαδικού τελεστή γίνεται ο έλεγχος και για τον τρίτο αριθμό, για να διαπιστωθεί αν αυτός είναι ο μέγιστος. Τέλος, με την χρήση της «if» ελέγχεται ποιος αριθμός έχει καταχωρηθεί στην μεταβλητή «Max» και τυπώνεται η σειρά που εισήχθη. Ο έλεγχος γίνεται και για τις τρεις μεταβλητές που διαβάζονται από την «standard» είσοδο, καθώς ο ίδιος αριθμός μπορεί να έχει διαβαστεί πάνω από μία φορά.

## Πρόγραμμα «switch.c»

Το πρόγραμμα βασισμένο στην εντολή ελέγχου «switch» επιτυγχάνει να διασαφηνίσει την λειτουργία της εντολής «switch» και «break». Αναλυτικά, ελέγχει την τιμή της ακέραιας παράστασης «i» που είναι «7» και εκτελεί εντολές από την «case 3 + 4» (case 7), όπου τυπώνεται το μήνυμα «7» και ο χαρακτήρας διαφυγής της αλλαγής γραμμής «\n». Στη συνέχεια, εκτελεί από την «default» την εντολή, όπου τυπώνεται το μήνυμα «default» και ο χαρακτήρας διαφυγής της αλλαγής γραμμής «\n». Το ίδιο και από την «case 1» την εντολή, όπου τυπώνεται το μήνυμα «1» και ο χαρακτήρας διαφυγής της αλλαγής γραμμής «\n». Τέλος, εκτελείται και η «break;» αμέσως μετά την «case 1» και η ροή του προγράμματος μεταφέρεται εκτός του παραθύρου εντολών της «switch».

## ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ

### Θέμα 3

#### Επισημάνση «C1.c»

Το «Πρόγραμμα “C1.c”» (Πηγαίος Κώδικας) και η «Τεκμηρίωση “C1.c”» (Ζητούμενο, Δομή, Μεταβλητές, Διάσχιση, Παραδείγματα, Παρατηρήσεις) απαντούν στο ζητούμενο του «Θέμα 3»

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Πρόγραμμα «C1.c»

```
1  #include <stdio.h>
2  #include <math.h>
3
4  int main (int argc, int **argv)
5
6  {
7      system ("chcp 1253");
8      printf ("=====\n\n");
9      // Δήλωση μεταβλητών
10     double a, b, c;
11     double x;
12     double D;
13     double root_D;
14     double x1, x2;
15     double x1_2;
16     // Γενικός τύπος δευτεροβάθμιας εξίσωσης
17     printf ("a*x^2 + b*x + c = 0\n\n");
18
19     printf ("=====\n\n");
20     // Εισαγωγή των συντελεστών της δευτεροβάθμιας εξίσωσης
21     printf ("Εισάγετε τον συντελεστή a : ");
22     scanf ("%lf", &a);
23     printf ("Εισάγετε τον συντελεστή b : ");
24     scanf ("%lf", &b);
25     printf ("Εισάγετε τον συντελεστή c : ");
26     scanf ("%lf", &c);
27     printf ("\n");
28
29     if (a != 0) // Έλεγχος των συντελεστών για την εύρεση και τον
30                υπολογισμό πραγματικών ριζών
31     {
32         // a != 0
33         D = (b * b) - 4 * a * c; // Διακρίνουσα
34
35         printf ("=====\n\n");
36         printf ("Η εξίσωση είναι δευτεροβάθμια.\n\n");
37         printf ("Η διακρίνουσα είναι : %lf\n\n", D);
38         printf ("=====\n\n");
39         if (D >= 0) // Έλεγχος διακρίνουσας
40         {
41             if (D == 0) // Διακρίνουσα ίση με 0
```



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
34         {
35             x1_2 = - b / (2 * a);
36             printf ("Η εξίσωση είναι αόριστη.\n");
37             printf ("Η πραγματική ρίζα x1_2 είναι διπλή.\n");
38             printf ("\n");
39             printf ("Ρίζα x1_2 : %lf\n", x1_2);
40         }
41     else // Διακρίνουσα θετική
42     {
43         root_D = sqrt ((float) D);
44         x1 = (- b + root_D) / (2 * a);
45         x2 = (- b - root_D) / (2 * a);
46         printf ("Πλήθος πραγματικών ριζών : 2\n\n");
47         printf ("Ρίζα x1 : %lf\n", x1);
48         printf ("Ρίζα x2 : %lf\n", x2);
49     }
50 }
51 else // Διακρίνουσα αρνητική
52 {
53     printf ("Η εξίσωση είναι αδύνατη.\n\n");
54     printf ("Πλήθος πραγματικών ριζών : 0\n");
55 }
56 }
57 else // a == 0
58 {
59     x = - c / b;
60     printf ("=====\n\n");
61     printf ("Η εξίσωση είναι πρωτοβάθμια.\n\n");
62     printf ("=====\n\n");
63     printf ("Πλήθος πραγματικών ριζών : 1\n\n");
64     printf ("Ρίζα x : %lf\n", x);
65 }
66
67 return 0;
68
69 }
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Τεκμηρίωση «C1.c»

### **Ζητούμενο**

Το πρόγραμμα «C1.c» επιτυγχάνει τις εξής λειτουργίες:

- i) Διαβάζει από την «standard» είσοδο τους τρεις συντελεστές μίας δευτεροβάθμιας εξίσωσης στο σύνολο των πραγματικών (τύπου double).
- ii) Ελέγχει τα περιεχόμενα των συντελεστών.
- iii) Υπολογίζει τις ρίζες της εξίσωσης ανάλογα τις τιμές που έχουν καταχωρηθεί στους συντελεστές.
- iv) Τυπώνει τα κατάλληλα μηνύματα (για τον τύπο της εξίσωσης και το πλήθος των ριζών της).

### **Δομή**

Προκειμένου να υλοποιηθεί το ζητούμενο χρησιμοποιήθηκαν, αρχικά, οι βιβλιοθήκες :

- i) «stdio.h» που περιέχει τις έτοιμες συναρτήσεις «scanf()» και «printf()» που συνδέονται με τα κανάλια εισόδου και εξόδου αντίστοιχα για την ανάγνωση και τηντύπωση περιεχομένων των αντίστοιχων μεταβλητών. Επίσης, η «printf()» χρησιμοποιήθηκε για να τυπωθούν χαρακτηριστικά μηνύματα για την βέλτιστη κατανόηση του πηγαίου κώδικα.
- ii) «math.h» που περιέχει την έτοιμη μαθηματική συνάρτηση «sqrt()» για τον υπολογισμό της διακρίνουσας της δευτεροβάθμιας εξίσωσης.

Επιπρόσθετα, χρησιμοποιήθηκαν οι χαρακτηριστικοί τελεστές :

- i) αριθμητικοί : +, -, \*, /
- ii) σχεσιακοί : !=, >=, ==
- iii) ανάθεσης : =
- iv) τελεστής & για την διεύθυνση μεταβλητής στην «scanf()»

Οι εντολές ελέγχου :

- i) if - else

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Μεταβλητές

Πραγματικές μεταβλητές (τύπου double) :

a (Ο συντελεστής «a» της δευτεροβάθμιας εξίσωσης)

b (Ο συντελεστής «b» της δευτεροβάθμιας εξίσωσης)

c (Ο συντελεστής «c» της δευτεροβάθμιας εξίσωσης)

x (Η ρίζα της εξίσωσης όταν «a = 0»)

D (Η διακρίνουσα)

root\_D (Η ρίζα της διακρίνουσας)

x1 (Η πρώτη ρίζα της εξίσωσης, όταν «a != 0» και «D > 0»)

x2 (Η δεύτερη ρίζα της εξίσωσης, όταν «a != 0» και «D > 0»)

x1\_2 (Η διπλή ρίζα της εξίσωσης, όταν «a != 0» και «D == 0»)

## Διάσχιση

Δήλωση μεταβλητών (γραμμές 9-14) :

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στις γραμμές «9-14» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Γενικός τύπος δευτεροβάθμιας εξίσωσης (γραμμές 15) :

Στην γραμμή «15» με μία συνάρτηση «printf()» τυπώνεται από την «standard» έξοδο το μήνυμα «a\*x^2 + b\*x + c = 0\n\n». Πιο αναλυτικά, τυπώνεται ο γενικός τύπος της δευτεροβάθμιας εξίσωσης και δύο χαρακτηριστικές διαφυγής αλλαγής γραμμής (\n).

Εισαγωγή των συντελεστών της δευτεροβάθμιας εξίσωσης (γραμμές 17-22) :

Στη συνέχεια, στις γραμμές «17-22» με την συνάρτηση «scanf()» διαβάζει από την «standard» είσοδο τους τρεις συντελεστές της εξίσωσης (γραμμές 18, 20, 22) και με την συνάρτηση «printf()» τυπώνονται από την «standard» έξοδο τα κατάλληλα μηνύματα για την εισαγωγή τους (γραμμές 17, 19, 21).

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Έλεγχος του συντελεστή «a» για την εύρεση και τον υπολογισμό πραγματικών ριζών (γραμμές 24-65) :

Στην γραμμή «24-65» υλοποιείται ένας πλήρης και αναλυτικός έλεγχος της εξίσωσης για την εύρεση και τον υπολογισμό πιθανών πραγματικών ριζών. Ο πιο σημαντικός παράγοντας είναι η τιμή που θα πάρει ο συντελεστής «a», καθώς αυτός καθορίζει τι βαθμού είναι η εξίσωση που υπολογίζουμε τις πραγματικές ρίζες της (πρωτοβάθμια ή δευτεροβάθμια). Στην γραμμή «24» με μία εντολή «if» ελέγχουμε αν «a != 0» και στην περίπτωση που η παράσταση παράγει αποτέλεσμα την τιμή «1» (δηλαδή, ο «a» δεν περιέχει την τιμή «0»), τότε εκτελούνται οι εντολές στις γραμμές «25-54». Διαφορετικά, δηλαδή, όταν η παράσταση παράγει αποτέλεσμα την τιμή «0», (ο «a» περιέχει την τιμή «0») εκτελούνται οι εντολές στις γραμμές «58-64», που είναι ενσωματωμένες σε μία εντολή «else» που αντιστοιχεί στην «if» της γραμμής «24».

**a != 0 (γραμμές 25-55) :**

Στις γραμμές «25-55» εκτελούνται οι εντολές για την περίπτωση που η παράσταση στην «if» ( $a \neq 0$ ) της γραμμής «25» παράγει αποτέλεσμα την τιμή «1» (ο συντελεστής «a» δεν περιέχει την τιμή «0»).

**Διακρίνουσα (γραμμές 26-30) :**

Στις γραμμές «26-30» συντάσσεται ο τύπος της διακρίνουσας της δευτεροβάθμιας εξίσωσης (γραμμή 26). Με την συνάρτηση «printf()» τυπώνεται το κατάλληλο μήνυμα από την «standard» έξοδο για να διασαφηνιστεί ότι η εξίσωση είναι δευτεροβάθμια (γραμμή 28), καθώς και το περιεχόμενο της διακρίνουσας στο σύνολο των πραγματικών αριθμών συνοδευόμενο με τα κατάλληλα μηνύματα (γραμμή 29).

**Έλεγχος διακρίνουσας (γραμμές 31-55) :**

Στις γραμμές «31-55» υλοποιείται ένας πλήρης και αναλυτικός έλεγχος του προσήμου της διακρίνουσας. Ανάλογα, το πρόσημο της διακρίνουσας καθορίζεται και το πλήθος των ριζών της εξίσωσης. Με μια εντολή «if-else» ελέγχεται αν η παράσταση «D >= 0» (αν η διακρίνουσα είναι μεγαλύτερη ή ίση του «0») παράγει αποτέλεσμα την τιμή «1», ώστε να εκτελεστούν οι εντολές στις γραμμές «32-49» ή αποτέλεσμα την τιμή «0», ώστε να εκτελεστούν οι εντολές στις γραμμές «51-55».

**Διακρίνουσα ίση με 0 (γραμμές 33-40) :**

Οι εντολές στις γραμμές «33-40» εκτελούνται όταν η παράσταση μέσα στην «if» της γραμμής «31» ( $D \geq 0$ ) παράγει αποτέλεσμα «1», δηλαδή, η διακρίνουσα είναι μεγαλύτερη ή ίση με το 0 και όταν η παράσταση μέσα

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

στην «if» της γραμμής «33» ( $D == 0$ ) παράγει εξίσου αποτέλεσμα «1», δηλαδή, η διακρίνουσα είναι ίση με το «0». Πιο αναλυτικά, καταχωρείται στην μεταβλητή «x1\_2» η παράσταση « $-b / (2 * a)$ » που είναι το τύπος για τον υπολογισμό της πραγματικής ρίζας στην περίπτωση που η διακρίνουσα είναι ίση με «0». Με την συνάρτηση «printf()» αποτυπώνονται από την «standard» έξοδο τα κατάλληλα μηνύματα για την εξίσωση και το πλήθος των πραγματικών ριζών της (γραμμές 36, 37), καθώς και το περιεχόμενο της μεταβλητής «x1\_2» που είναι η πραγματική ρίζα της εξίσωσης (γραμμή 39).

## Διακρίνουσα θετική (γραμμές 41-49) :

Οι εντολές στις γραμμές «41-49» εκτελούνται όταν η παράσταση μέσα στην «if» της γραμμής «31» ( $D \geq 0$ ) παράγει αποτέλεσμα «1», δηλαδή, η διακρίνουσα είναι μεγαλύτερη ή ίση με το «0» και όταν η παράσταση μέσα στην «if» της γραμμής «33» ( $D == 0$ ) παράγει αποτέλεσμα «0», δηλαδή, η διακρίνουσα δεν είναι ίση με το «0». Πιο αναλυτικά, έχουμε την περίπτωση της θετικής διακρίνουσας και εκτελούνται οι εντολές που είναι ενσωματωμένες στην «else» της γραμμής «41». Καταρχήν, με την συνάρτηση «sqrt()» που εισάγουμε έτοιμη από την βιβλιοθήκη «math.h» καταχωρείται στην μεταβλητή «root\_D» η ρίζα της διακρίνουσας (γραμμή 43). Στις μεταβλητές «x1» και «x2» καταχωρούνται οι τύποι για τον υπολογισμό των ριζών της εξίσωσης στην περίπτωση που η διακρίνουσα είναι θετική (γραμμές 44, 45). Με την συνάρτηση «printf()» αποτυπώνονται από την «standard» έξοδο τα κατάλληλα μηνύματα για την εξίσωση και το πλήθος των πραγματικών ριζών της (γραμμή 46), καθώς και τα περιεχόμενα των μεταβλητών «x1», «x2» που είναι οι πραγματικές ρίζες της εξίσωσης (γραμμές 47, 48).

## Διακρίνουσα αρνητική (γραμμές 51-55) :

Οι εντολές στις γραμμές «51-55» εκτελούνται ,εφόσον, η παράσταση μέσα στην «if» της γραμμής «31» ( $D \geq 0$ ) παράγει αποτέλεσμα «0», δηλαδή, η διακρίνουσα δεν είναι μεγαλύτερη ή ίση με το «0». Πιο αναλυτικά, έχουμε την περίπτωση της αρνητικής διακρίνουσας και εκτελούνται οι εντολές που είναι ενσωματωμένες στην «else» της γραμμής «51». Η εντολή αυτή περιλαμβάνει δύο συναρτήσεις «printf()» που τυπώνουν από την «standard» έξοδο δύο μηνύματα για τον τύπο της εξίσωσης και το πλήθος των πραγματικών ριζών της (γραμμές 53, 54).

## **a == 0 (γραμμές 57-65) :**

Οι εντολές στις γραμμές «57-65» εκτελούνται ,εφόσον, η παράσταση ( $a != 0$ ) στην «if» της γραμμής «24» παράγει αποτέλεσμα την τιμή «0» (ο συντελεστής περιέχει την τιμή «0»). Η «if» αυτή αντιστοιχεί σε μία

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

«else» στην γραμμή «57», οπότε θα εκτελεστούν οι εντολές που είναι ενσωματωμένες στην δεύτερη αυτή εντολή. Η «else» περιέχει την μεταβλητή «x» και την παράσταση που καταχωρείται, όπου αποτελεί την πραγματική ρίζα της εξίσωσης (γραμμή 59), καθώς και συναρτήσεις «printf()» που τυπώνουν από την «standard» έξοδο τα κατάλληλα μηνύματα για τον τύπο της εξίσωσης, το πλήθος των ριζών της (γραμμές 61, 63) και το περιεχόμενο της μεταβλητής «x» (γραμμή 64).

## Παραδείγματα

### Παράδειγμα 1 (a == 0)

Active code page: 1253

=====

$$a*x^2 + b*x + c = 0$$

=====

Εισάγετε τον συντελεστή a : 0.0

Εισάγετε τον συντελεστή b : -11.0

Εισάγετε τον συντελεστή c : 3.0

=====

Η εξίσωση είναι πρωτοβάθμια.

=====

Πλήθος πραγματικών ριζών : 1

Ρίζα x : 0.272727

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Παράδειγμα 2 (a != 0 / D >= 0 / D == 0)

Active code page: 1253

=====

$$a*x^2 + b*x + c = 0$$

=====

Εισάγετε τον συντελεστή a : 2.0

Εισάγετε τον συντελεστή b : 4.0

Εισάγετε τον συντελεστή c : 2.0

=====

Η εξίσωση είναι δευτεροβάθμια.

Η διακρίνουσα είναι : 0.000000

=====

Η εξίσωση είναι αόριστη.

Η πραγματική ρίζα x1\_2 είναι διπλή.

Ρίζα x1\_2 : -1.000000

## Παράδειγμα 3 (a != 0 / D >= 0 / D != 0)

Active code page: 1253

=====

$$a*x^2 + b*x + c = 0$$

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

=====

Εισάγετε τον συντελεστή a : 2.0

Εισάγετε τον συντελεστή b : -7.0

Εισάγετε τον συντελεστή c : -1.0

=====

Η εξίσωση είναι δευτεροβάθμια.

Η διακρίνουσα είναι : 57.000000

=====

Πλήθος πραγματικών ριζών : 2

Ρίζα x1 : 3.637459

Ρίζα x2 : -0.137459

## **Παράδειγμα 4 (a != 0 / D < 0)**

Active code page: 1253

=====

$$a \cdot x^2 + b \cdot x + c = 0$$

=====

Εισάγετε τον συντελεστή a : 32.0

Εισάγετε τον συντελεστή b : 1.0

Εισάγετε τον συντελεστή c : 16.0



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

=====

Η εξίσωση είναι δευτεροβάθμια.

Η διακρίνουσα είναι : -2047.000000

=====

Η εξίσωση είναι αδύνατη.

Πλήθος πραγματικών ριζών : 0

## **Παρατηρήσεις**

Η ορθότητα των αποτελεσμάτων στα παραπάνω παραδείγματα ελέγχθηκε και με την χρήση τυπικής αριθμομηχανής και δεν διαπίστωθηκε καμία διαφορά.

Αναλυτικά, έχουμε τέσσερα παραδείγματα όπου το καθένα αντιστοιχεί σε ειδικές συνθήκες που βρίσκονται μέσα στις αντίστοιχες εντολές «if-else» :

### **Παράδειγμα 1 (a == 0) :**

Αντιπροσωπεύει την περίπτωση, όπου ο συντελεστής «a» της εξίσωσης περιέχει την τιμή «0».

### **Παράδειγμα 2 (a != 0 / D >= 0 / D == 0) :**

Αντιπροσωπεύει την περίπτωση, όπου ο συντελεστής «a» της εξίσωσης δεν περιέχει την τιμή «0» και η διακρίνουσα είναι ίση με το «0».

### **Παράδειγμα 3 (a != 0 / D >= 0 / D != 0) :**

Αντιπροσωπεύει την περίπτωση, όπου ο συντελεστής «a» της εξίσωσης δεν περιέχει την τιμή «0» και η διακρίνουσα είναι θετική.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Παράδειγμα 4 ( $a \neq 0 / D < 0$ ) :

Αντιπροσωπεύει την περίπτωση, όπου ο συντελεστής «a» της εξίσωσης δεν περιέχει την τιμή «0» και η διακρίνουσα είναι αρνητική.

## Θέμα 4

### Επισημάνση «C2.c» :

Το «Πρόγραμμα «C2.c»» (Πηγαίος Κώδικας) και η «Τεκμηρίωση «C2»» (Ζητούμενο, Δομή, Μεταβλητές, Διάσχιση, Παραδείγματα, Παρατηρήσεις) απαντούν στο ζητούμενο του «Θέμα 4».

### Πρόγραμμα «C2.c»

```
1 #include <stdio.h>
2
3 int main (int argc, int **argv)
4
5 {
6     system ("chcp 1253");
7     printf ("=====\n\n");
8     // Δήλωση μεταβλητών
9     int x, y, z;
10    printf ("Εύρεση μέγιστου αριθμού και σειράς που εισήχθη\n\n");
11    // Τίτλος προγράμματος
12    printf ("=====\n\n");
13    // Εισαγωγή τριών ακεραίων αριθμών
14    printf ("Εισάγετε τον 1ο ακεραίο αριθμό : ");
15    scanf ("%d", &x);
16    printf ("Εισάγετε τον 2ο ακεραίο αριθμό : ");
17    scanf ("%d", &y);
18    printf ("Εισάγετε τον 3ο ακεραίο αριθμό : ");
19    scanf ("%d", &z);
20    printf ("\n=====\n\n");
21    // Έλεγχοι των ακεραίων για την εύρεση του μεγίστου αριθμού και της
22    // σειράς που εισήχθη
```

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
18      (x > y && x > z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 1ος.", x) : printf ("\b");
// x > y KAI x > z

19      (x < y && y > z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 2ος.", y) : printf ("\b");
// x < y KAI y > z

20      (x < z && y < z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 3ος.", z) : printf ("\b");
// x < z KAI y < z

21      (x == y && x > z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 1ος και 2ος.", x) : printf ("\b");
// x == y > z

22      (x < y && y == z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 2ος και 3ος.", y) : printf ("\b");
// x < y == z

23      (x > y && x == z) ? printf ("Ο μεγαλύτερος αριθμός είναι ο %d
και εισήχθη 1ος και 3ος.", z) : printf ("\b");
// x > x == z

24      (x == y && y == z && x == z) ? printf ("Ο μεγαλύτερος αριθμός
είναι ο %d και εισήχθη 1ος, 2ος και 3ος.\n", x) : printf ("\n");
// x == y == z

25

26      return 0;

27 }
```

## Τεκμηρίωση «C2.c»

### **Ζητούμενο**

Το πρόγραμμα «C2.c» επιτυγχάνει τις εξής λειτουργίες:

- i) Διαβάζει από την «standard» είσοδο τρεις αριθμούς στο σύνολο των ακεραίων (τύπου int).
- ii) Συγκρίνει τους τρεις αριθμούς, οι οποίοι έχουν καταχωρηθεί στα περιεχόμενα τριών μεταβλητών.
- iii) Τυπώνει τα κατάλληλα μηνύματα που αποκαλύπτουν ποιος από τους τρεις αριθμούς είναι ο μεγαλύτερος, καθώς και με ποια σειρά εισήχθη ή εισήχθησαν στην περίπτωση που έχει καταχωρηθεί ο ίδιος αριθμός σε παραπάνω από μία μεταβλητές.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## **Δομή**

Προκειμένου να υλοποιηθεί το ζητούμενο χρησιμοποιήθηκαν, αρχικά, οι βιβλιοθήκες :

i) «stdio.h» που περιέχει τις έτοιμες συναρτήσεις «scanf()» και «printf()» που συνδέονται με τα κανάλια εισόδου και εξόδου αντίστοιχα για την ανάγνωση και τηντύπωση περιεχομένων των αντίστοιχων μεταβλητών. Επίσης, η «printf()» χρησιμοποιήθηκε για να τυπωθούν χαρακτηριστικά μηνύματα για την βέλτιστη κατανόηση του πηγαίου κώδικα.

Επιπρόσθετα, χρησιμοποιήθηκαν οι χαρακτηριστικοί τελεστές :

i) συνθήκης : {? :}

ii) σχεσιακοί : <, >, ==

iii) ανάθεσης : =

iv) τελεστής & για την διεύθυνση μεταβλητής στην «scanf()»

## **Μεταβλητές**

Ακέραιες μεταβλητές (τύπου int) :

x (ο πρώτος ακέραιος αριθμός)

y (ο δεύτερος ακέραιος αριθμός)

z (ο τρίτος ακέραιος αριθμός)

## **Διάσχιση**

Δήλωση μεταβλητών (γραμμή 8) :

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στην γραμμή «8» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Τίτλος προγράμματος (γραμμή 9) :

Στην γραμμή «9» με μία συναρτήση «printf()» τυπώνεται από την «standard» έξοδο ο τίτλος του προγράμματος (Εύρεση μεγίστου αριθμού και σειράς που εισήχθη), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Εισαγωγή τριών ακέραιων αριθμών (γραμμές 11-16) :

Στις γραμμές «11-16» διαβάζονται από την «standard» είσοδο με την συνάρτηση «scanf()» οι τρεις ακέραιοι αριθμοί, συνοδευόμενοι από τα κατάλληλα μηνύματα που τυπώνονται από την «standard» έξοδο με την συνάρτηση «printf()».

## Έλεγχοι των ακεραίων για την εύρεση του μεγίστου αριθμού και της σειράς που εισήχθη (γραμμές 17-24) :

Στις γραμμές «17-24» υλοποιούνται οι έλεγχοι των ακεραίων αριθμών για την εύρεση του μεγίστου αριθμού, καθώς και της σειράς που εισήχθη. Η σύνταξη του πηγαίου κώδικα βασίζεται αποκλειστικά και μόνο στον τριαδικό τελεστή συνθήκης (? :).

### $x > y$ ΚΑΙ $x > z$ (γραμμή 18) :

Στην γραμμή «18» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στην μεταβλητή «x» και εισήχθη πρώτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση « $x > y \ \&\& \ x > z$ » παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, η μεταβλητή «x» περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «y» και «z», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «x» συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση « $x > y \ \&\& \ x > z$ » παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως η μεταβλητή «x» δεν περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «y» και «z», τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτής της μίας θέσης πίσω (\b).

### $x < y$ ΚΑΙ $y > z$ (γραμμή 19) :

Στην γραμμή «19» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στην μεταβλητή «y» και εισήχθη δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση « $x < y \ \&\& \ y > z$ » παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, η μεταβλητή «y» περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «x» και «z», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «y» συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση « $x < y \ \&\& \ y > z$ » παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως η μεταβλητή «y» δεν περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «x» και «z»,

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της μίας θέσης πίσω (\b).

## x < z ΚΑΙ y < z (γραμμή 20) :

Στην γραμμή «20» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στην μεταβλητή «z» και εισήχθη τρίτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «x < z && y < z» παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, η μεταβλητή «z» περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «x» και «y», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «z» συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση «x < z && y < z» παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως η μεταβλητή «z» δεν περιέχει τιμή μεγαλύτερη από αυτές των μεταβλητών «x» και «y», τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της μίας θέσης πίσω (\b).

## x == y > z (γραμμή 21) :

Στην γραμμή «21» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «x» και «y» και εισήχθη πρώτος και δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «x == y && y > z» παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, οι μεταβλητές «x» και «y» περιέχουν την ίδια τιμή που είναι μεγαλύτερη από αυτήν της μεταβλητής «z», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «x» (ή της «y») συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση «x == y && y > z» παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως οι μεταβλητές «x» και «y» δεν περιέχουν τιμή μεγαλύτερη από αυτήν της μεταβλητής «x» και «y», τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της μίας θέσης πίσω (\b).

## x < y == z (γραμμή 22) :

Στην γραμμή «22» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «y» και «z» και εισήχθη δεύτερος και τρίτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «x < y && y == z» παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, οι μεταβλητές «y» και «z» περιέχουν την ίδια τιμή που είναι μεγαλύτερη από αυτήν της μεταβλητής

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

«x», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «y» (ή της «z») συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση «x < y && y == z» παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως οι μεταβλητές «y» και «z» δεν περιέχουν τιμή μεγαλύτερη από αυτήν της μεταβλητής «x», τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της μίας θέσης πίσω (\b).

x == z > y (γραμμή 23) :

Στην γραμμή «23» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «x» και «z» και εισήχθη πρώτος και δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «x > y && x == z» παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, οι μεταβλητές «x» και «z» περιέχουν την ίδια τιμή που είναι μεγαλύτερη από αυτήν της μεταβλητής «y», τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «z» (ή της «x») συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση «x > y && x == z» παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως οι μεταβλητές «x» και «z» δεν περιέχουν τιμή μεγαλύτερη από αυτήν της μεταβλητής «y», τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της μίας θέσης πίσω (\b).

x == y == z (γραμμή 24) :

Στην γραμμή «24» συντάσσεται η περίπτωση, όπου ο ίδιος αριθμός καταχωρείται και στις τρεις μεταβλητές «x», «y», «z» και εισήχθη πρώτος, δεύτερος και τρίτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «x == y && y == z && x == z» παράγει αποτέλεσμα «True» (τιμή διάφορη του «0»), δηλαδή, οι μεταβλητές «x», «y» και «z» περιέχουν την ίδια τιμή, τότε τυπώνεται με την «printf()» από την «standard» έξοδο το περιεχόμενο της «x» (ή της «y» ή της «z») συνοδευόμενο με το κατάλληλο μήνυμα και για την σειρά που εισήχθη. Σε αντίθετη περίπτωση, δηλαδή, η παράσταση «x == y && y == z && x == z» παράγει αποτέλεσμα «False» (τιμή ίση με το «0») και επομένως οι μεταβλητές «x», «y» και «z» δεν περιέχουν την ίδια τιμή, τότε τυπώνεται με την «printf()» από την «standard» έξοδο ο χαρακτήρας διαφυγής αυτός της αλλαγής γραμμής (\n).

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## Παραδείγματα

### Παράδειγμα 1 ( $x > y$ ΚΑΙ $x > z / 1^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 8

Εισάγετε τον 2ο ακέραιο αριθμό : -2

Εισάγετε τον 3ο ακέραιο αριθμό : 0

=====

Ο μεγαλύτερος αριθμός είναι ο 8 και εισήχθη 1ος.

### Παράδειγμα 2 ( $x < y$ ΚΑΙ $y > z / 2^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 2

Εισάγετε τον 2ο ακέραιο αριθμό : 5

Εισάγετε τον 3ο ακέραιο αριθμό : 4

=====



# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο μεγαλύτερος αριθμός είναι ο 5 και εισήχθη 2ος.

## Παράδειγμα 3 ( $x < z$ ΚΑΙ $y < z / 3^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : -15

Εισάγετε τον 2ο ακέραιο αριθμό : -13

Εισάγετε τον 3ο ακέραιο αριθμό : -9

=====

Ο μεγαλύτερος αριθμός είναι ο -9 και εισήχθη 3ος.

## Παράδειγμα 4 ( $x == y > z / 1^{\text{ος}}, 2^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 32

Εισάγετε τον 2ο ακέραιο αριθμό : 32

Εισάγετε τον 3ο ακέραιο αριθμό : 19

=====

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο μεγαλύτερος αριθμός είναι ο 32 και εισήχθη 1ος και 2ος.

## Παράδειγμα 5 ( $x < y == z / 2^{\text{ος}}, 3^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : -20

Εισάγετε τον 2ο ακέραιο αριθμό : -4

Εισάγετε τον 3ο ακέραιο αριθμό : -4

=====

Ο μεγαλύτερος αριθμός είναι ο -4 και εισήχθη 2ος και 3ος.

## Παράδειγμα 6 ( $x == z > y / 1^{\text{ος}}, 3^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 8

Εισάγετε τον 2ο ακέραιο αριθμό : 4

Εισάγετε τον 3ο ακέραιο αριθμό : 8

=====

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο μεγαλύτερος αριθμός είναι ο 8 και εισήχθη 1ος και 3ος.

## Παράδειγμα 7 ( $x == y == z / 1^{\text{ος}}, 2^{\text{ος}}, 3^{\text{ος}}$ )

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 100

Εισάγετε τον 2ο ακέραιο αριθμό : 100

Εισάγετε τον 3ο ακέραιο αριθμό : 100

=====

Ο μεγαλύτερος αριθμός είναι ο 100 και εισήχθη 1ος, 2ος και 3ος.

## Παρατηρήσεις

Αναλυτικά, έχουμε επτά παραδείγματα όπου το καθένα αντιστοιχεί σε ειδικές συνθήκες που βρίσκονται μέσα στους αντίστοιχους τριαδικούς τελεστές συνθήκης «? :» :

## Παράδειγμα 1 ( $x > y$ ΚΑΙ $x > z / 1^{\text{ος}}$ ) :

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «x» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «y» και «z» και εισήχθη πρώτος.

## Παράδειγμα 2 ( $x < y$ ΚΑΙ $y > z / 2^{\text{ος}}$ ) :

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «y» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «x» και «z» και εισήχθη δεύτερος.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

## **Παράδειγμα 3 ( $x < z$ ΚΑΙ $y < z / 3^{\text{ος}}$ ) :**

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «z» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «x» και «y» και εισήχθη τρίτος.

## **Παράδειγμα 4 ( $x == y > z / 1^{\text{ος}}, 2^{\text{ος}}$ ) :**

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «x» και «y» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «z» και εισήχθη πρώτος και δεύτερος.

## **Παράδειγμα 5 ( $x < y == z / 2^{\text{ος}}, 3^{\text{ος}}$ ) :**

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «y» και «z» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «x» και εισήχθη δεύτερος και τρίτος.

## **Παράδειγμα 6 ( $x == z > y / 1^{\text{ος}}, 3^{\text{ος}}$ ) :**

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «x» και «z» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «y» και εισήχθη πρώτος και τρίτος.

## **Παράδειγμα 7 ( $x == y == z / 1^{\text{ος}}, 2^{\text{ος}}, 3^{\text{ος}}$ ) :**

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «x», «y» και «z» περιέχουν τον ίδιο και εισήχθη πρώτος, δεύτερος και τρίτος.

# ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



Σας ευχαριστώ για την προσοχή σας.

