

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

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ (ΣΕΛΙΔΕΣ 3-5)

ΘΕΜΑ 1	(ΣΕΛΙΔΕΣ 3-5)
ΘΕΜΑ 1a) Τι γνωρίζετε για την εντολή «if»;	(ΣΕΛΙΔΕΣ 3-4)
ΘΕΜΑ 1b) Τι γνωρίζετε για την εντολή «switch»;	(ΣΕΛΙΔΕΣ 4-5)

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΣΧΕΤΙΚΑ ΜΕ ΤΟΥΣ ΠΗΓΑΙΟΥΣ ΚΩΔΙΚΕΣ (ΣΕΛΙΔΕΣ 5-6)

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

ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ (ΣΕΛΙΔΕΣ 7-35)

ΘΕΜΑ 3	(ΣΕΛΙΔΕΣ 6-21)
Επισήμανση «2ndGradeEquation.c»	(ΣΕΛΙΔΑ 6)
Πρόγραμμα «2ndGradeEquation.c»	(ΣΕΛΙΔΕΣ 7-9)
Τεκμηρίωση «2ndGradeEquation.c»	(ΣΕΛΙΔΕΣ 9-20)
{	
Ζητούμενο	(ΣΕΛΙΔΑ 9)
Δομή	(ΣΕΛΙΔΕΣ 9-10)
Μεταβλητές	(ΣΕΛΙΔΕΣ 10-11)
Διάσχιση	(ΣΕΛΙΔΕΣ 11-16)
Παραδείγματα	(ΣΕΛΙΔΕΣ 16-19)
Παρατηρήσεις	(ΣΕΛΙΔΕΣ 19-21)
}	

ΘΕΜΑ 4	(ΣΕΛΙΔΕΣ 22-35)
Επισήμανση «MaxInteger.c»	(ΣΕΛΙΔΑ 22)
Πρόγραμμα «MaxInteger.c»	(ΣΕΛΙΔΕΣ 22-23)
Τεκμηρίωση «MaxInteger.c»	(ΣΕΛΙΔΕΣ 23-35)
{	
Ζητούμενο	(ΣΕΛΙΔΑ 23)
Δομή	(ΣΕΛΙΔΕΣ 23-24)
Μεταβλητές	(ΣΕΛΙΔΑ 24)
Διάσχιση	(ΣΕΛΙΔΕΣ 24-28)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Παραδείγματα
Παρατηρήσεις

(ΣΕΛΙΔΕΣ 28-33)
(ΣΕΛΙΔΕΣ 33-35)

}

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ

ΘΕΜΑ 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 «True», δηλαδή, 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 «False» (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 μία τιμή «False» (0) και μία «True» (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 «True» (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 «False» (0), τότε θα εκτελεστούν οι εντολές που είναι μέσα στην «else» και στην συγκεκριμένη περίπτωση είναι η «printf ("Goodbye wolrd");», όπου θα τυπωθεί το μήνυμα «Goodbye world;».

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

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

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» («False» ή «True» αντίστοιχα). Αν παράγει αποτέλεσμα διάφορο του «0» (True), 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» («False» ή «True»). Αν παράγει αποτέλεσμα διάφορο του «0» (True), 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». Ειδικά, 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 «0» (False), 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

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

γραμμαής «\n». Σε περίπτωση που η μεταβλητή περιέχει την τιμή «2», τότε εκτελούνται οι εντολές στην «case 2» και στη «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 «default».

Επιπρόσθετα, υπάρχει η εντολή «break» που ουσιαστικά διακόπτει την ροή του προγράμματος μέσα στο παράθυρο εντολών της «switch» και ο έλεγχος μεταφέρεται έξω από αυτό. 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

ΘΕΜΑ 2

ΕΠΙΣΗΜΑΝΣΗ «if.c» / «if1.c» / switch.c

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

ΤΕΚΜΗΡΙΩΣΗ «if.c»

Το πρόγραμμα «if.c» βασισμένο σε εντολές ελέγχου «if-else» διαβάζει από τη «standard» είσοδο τρεις ακέραιους 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

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

εντολών «if-else».

ΤΕΚΜΗΡΙΩΣΗ «if1.c»

Το πρόγραμμα «if1.c» βασισμένο στην εντολή ελέγχου «if» και του 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 πρώτο 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 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 τη «standard» είσοδο, καθώς ο ίδιος αριθμός μπορεί να έχει διαβαστεί πάνω από μία φορά.

ΤΕΚΜΗΡΙΩΣΗ «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 μεταβλητής «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

ΘΕΜΑ 3

ΕΠΙΣΗΜΑΝΣΗ «2ndGradeEquation.c»

Το «Πρόγραμμα “2ndGradeEquation.c”» (Πηγαίος Κώδικας) και η «Τεκμηρίωση “2ndGrade.c”» (Ζητούμενο, Δομή, Μεταβλητές, Διάσχιση, Παραδείγματα, Παρατηρήσεις) απαντούν στο ζητούμενο του «Θέμα 3»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΡΟΓΡΑΜΜΑ «2ndGradeEquation.c»

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main (int argc, char **argv)
5
6 {
7
8     system ("chcp 1253");
9
10    double a, b, c; // Δήλωση μεταβλητών
11    double x;
12    double D;
13    double root_D;
14    double x1, x2;
15    double x1_2;
16
17    printf ("=====\n\n");
18    printf ("Επίλυση δευτεροβάθμιας εξίσωσης\n\n"); // Τίτλος του
    προγράμματος
19    printf ("=====\n\n");
20    printf ("Εισάγετε τον συντελεστή a : ");
21    scanf ("%lf", &a); // Εισαγωγή του 1ου συντελεστή "a" της
    εξίσωσης
22    printf ("Εισάγετε τον συντελεστή b : ");
23    scanf ("%lf", &b); // Εισαγωγή του 2ου συντελεστή "b" της
    εξίσωσης
24    printf ("Εισάγετε τον συντελεστή c : ");
25    scanf ("%lf", &c); // Εισαγωγή του 3ου συντελεστή "c" της
    εξίσωσης
26    printf ("\n-----\n\n");
27    printf ("Συντελεστής a : [%20.6lf]\n", a); // Εκτύπωση του 1ου
    συντελεστή "a" της εξίσωσης
28    printf ("Συντελεστής b : [%20.6lf]\n", b); // Εκτύπωση του 2ου
    συντελεστή "b" της εξίσωσης
29    printf ("Συντελεστής c : [%20.6lf]\n\n", c); // Εκτύπωση του
    3ου συντελεστή "c" της εξίσωσης
```


ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
30     if (a != 0) /* (~) Δευτεροβάθμια εξίσωση */
31     {
32         D = (b * b) - 4 * a * c; // Υπολογισμός της διακρίνουσας
33         printf ("Η εξίσωση είναι δευτεροβάθμια\n\n");
34         printf ("Διακρίνουσα : [%20.6lf]\n\n", D); // Εκτύπωση
           της διακρίνουσας
35         if (D >= 0) /* (!) Θετική διακρίνουσα ή ίση με "0" */
36         {
37             if (D == 0) /* (+) Διακρίνουσα ίση με "0" */
38             {
39                 x1_2 = - b / (2 * a); // Υπολογισμός της
           διπλής πραγματικής ρίζας
40                 printf ("Η εξίσωση είναι αόριστη\n");
41                 printf ("Η πραγματική ρίζα x1_2 είναι
           διπλή\n\n");
42                 printf ("Ρίζα x1_2 : [%20.6lf]\n\n",
           x1_2); // Εκτύπωση της διπλής πραγματικής ρίζας
43                 printf ("-----\n\n");
44             }
45             else /* (+) Διακρίνουσα θετική */
46             {
47                 root_D = sqrt (D); // Υπολογισμός της
           τετραγωνικής ρίζας της διακρίνουσας
48                 x1 = (- b + root_D) / (2 * a);
           // Υπολογισμός της πρώτης πραγματικής ρίζας
49                 x2 = (- b - root_D) / (2 * a);
           // Υπολογισμός της δεύτερης πραγματικής ρίζας
50                 printf ("Πλήθος πραγματικών ριζών :
           2\n\n");
51                 printf ("Ρίζα x1 : [%20.6lf]\n", x1);
           // Εκτύπωση της πρώτης πραγματικής ρίζας
52                 printf ("Ρίζα x2 : [%20.6lf]\n\n",
           x2); // Εκτύπωση της δεύτερης πραγματικής ρίζας
53                 printf ("-----\n\n");
54             }
55         }
56         else /* (!) Αρνητική διακρίνουσα */
57         {
58             printf ("Η εξίσωση είναι αδύνατη\n\n");
59             printf ("Πλήθος πραγματικών ριζών : 0\n\n");
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
60          printf ("-----\n\n");
-----\n\n");
61      }
62  }
63  else /* (~) Πρωτοβάθμια εξίσωση */
64  {
65      x = - c / b; // Υπολογισμός της πραγματικής ρίζας
66      printf ("Η εξίσωση είναι πρωτοβάθμια\n");
67      printf ("Πλήθος πραγματικών ριζών : 1\n\n");
68      printf ("Ρίζα x          : [%20.6lf]\n\n", x); // Εκτύπωση
της πραγματικής ρίζας
69      printf ("-----\n\n");
\n\n");
70  }
71
72  return 0;
73
74 }
```

ΤΕΚΜΗΡΙΩΣΗ «2ndGradeEquation.c»

ΖΗΤΟΥΜΕΝΟ

Το πρόγραμμα «2ndGradeEquation.c» επιτυγχάνει τις εξής λειτουργίες:

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

ΔΟΜΗ

Προκειμένου να υλοποιηθεί το ζητούμενο χρησιμοποιήθηκαν, αρχικά,

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

οι βιβλιοθήκες :

a) «stdio.h» : 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.

b) «math.h» : Περιέχει την έτοιμη μαθηματική συνάρτηση «sqrt()» για τον υπολογισμό της διακρίνουσας της δευτεροβάθμιας εξίσωσης.

Επιπρόσθετα, χρησιμοποιήθηκαν οι χαρακτηριστικοί τελεστές :

a) αριθμητικοί : +, -, *, /

b) σχεσιακοί : !=, >=, ==

c) ανάθεσης : =

d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης της «standard» εισόδου, «scanf()».

Οι εντολές ελέγχου :

a) if - else

ΜΕΤΑΒΛΗΤΕΣ

Πραγματικές μεταβλητές διπλής ακρίβειας (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»)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

x1_2 (Η διπλή ρίζα της εξίσωσης, όταν «a != 0» και «D == 0»)

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών

(γραμμές 10-15)

Το πρόγραμμα ξεκινάει με τη δήλωση των μεταβλητών στις γραμμές «10-15» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Τίτλος του προγράμματος

(γραμμή 18)

Στη γραμμή «18» με μία συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο ο τίτλος του προγράμματος ($a \cdot x^2 + b \cdot x + c = 0$), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

Εισαγωγή του 1ου συντελεστή "a" της εξίσωσης

(γραμμή 21)

Στη γραμμή «21» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο πρώτος πραγματικός συντελεστής «a» της εξίσωσης. Με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 20).

Εισαγωγή του 2ου συντελεστή "b" της εξίσωσης

(γραμμή 23)

Στη γραμμή «23» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο δεύτερος πραγματικός συντελεστής «b» της εξίσωσης. Στη συνέχεια, με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 22).

Εισαγωγή του 3ου συντελεστή "c" της εξίσωσης

(γραμμή 25)

Στη γραμμή «25» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο τρίτος πραγματικός συντελεστής «c» της εξίσωσης. Στη συνέχεια, με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 24).

Εκτύπωση του 1ου συντελεστή "a" της εξίσωσης

(γραμμή 27)

Στη γραμμή «27» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «a» (ο πρώτος συντελεστής της εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των απο-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

τελεσμάτων.

Εκτύπωση του 2ου συντελεστή "b" της εξίσωσης (γραμμή 28)

Στη γραμμή «28» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «b» (ο δεύτερος συντελεστής της εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του 3ου συντελεστή "c" της εξίσωσης (γραμμή 29)

Στη γραμμή «29» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «c» (ο τρίτος συντελεστής της εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

(~) Δευτεροβάθμια εξίσωση (γραμμές 30-62) **(~) Πρωτοβάθμια εξίσωση (γραμμές 63-70)**

Στις γραμμές «30-70» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για το περιεχόμενο του πρώτου συντελεστή «a» της εξίσωσης.

Διεξοδικά, αν ο «a» δεν περιέχει τη τιμή «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Δευτεροβάθμια εξίσωση (γραμμές 30-62)», διαφορετικά αν περιέχει τη τιμή «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Πρωτοβάθμια εξίσωση (γραμμές 63-70)».

(~) Δευτεροβάθμια εξίσωση (γραμμές 30-62)

Στις γραμμές «30-62» εκτελούνται οι εντολές για την περίπτωση που η παράσταση στην «if» ($a \neq 0$) της γραμμής «30» παράγει αποτέλεσμα μία τιμή «True» (ο συντελεστής «a» δεν περιέχει την τιμή «0»).

Υπολογισμός της διακρίνουσας (γραμμή 32)

Στη γραμμή «32» υπολογίζεται με την αριθμητική παράσταση $(b * b - 4 * a * c)$ η διακρίνουσα για τον υπολογισμό των πραγματικών ριζών της δευτεροβάθμιας εξίσωσης και το αποτέλεσμα καταχωρείται στη με-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ταβλητή «D».

Εκτύπωση της διακρίνουσας **(γραμμή 34)**

Στη γραμμή «35» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «D» (η διακρίνουσα) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

(!) Θετική διακρίνουσα ή ίση με "0" **(γραμμές 35-56)** **(!) Αρνητική διακρίνουσα** **(γραμμές 56-61)**

Στις γραμμές «35-61» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για το περιεχόμενο της διακρίνουσας (μεταβλητή «D»).

Διεξοδικά, αν η «D» περιέχει τιμή μεγαλύτερη ή ίση με το «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Θετική διακρίνουσα ή ίση με "0" (γραμμές 35-56)», διαφορετικά αν περιέχει τιμή μικρότερη του «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Αρνητική διακρίνουσα (γραμμές 56-61)».

(!) Θετική διακρίνουσα ή ίση με "0" **(γραμμές 35-56)**

Στις γραμμές «35-56» εκτελούνται οι εντολές για την περίπτωση που η παράσταση στην «if» ($D \geq 0$) της γραμμής «35» παράγει αποτέλεσμα μία τιμή «True» (η διακρίνουσα «D» περιέχει τιμή ίση ή μεγαλύτερη του «0»).

(+) Διακρίνουσα ίση με "0" **(γραμμές 37-44)** **(+) Διακρίνουσα θετική** **(γραμμές 45-54)**

Στις γραμμές «37-54» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για το περιεχόμενο της διακρίνουσας (μεταβλητή «D»).

Διεξοδικά, αν η «D» περιέχει τιμή ίση με το «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Διακρίνουσα ίση με "0" (γραμμές 37-44)», διαφορετικά αν δεν περιέχει τιμή ίση με το «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Διακρίνουσα θετική (γραμμές 45-54)».

(+) Διακρίνουσα ίση με "0" **(γραμμές 37-44)**

Στις γραμμές «37-44» εκτελούνται οι εντολές για την περίπτωση που η παράσταση στην «if» ($D == 0$) της γραμμής «37» παράγει αποτέλεσμα μία τιμή «True» (η διακρίνουσα «D» περιέχει τιμή ίση με το «0»).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Υπολογισμός της διπλής πραγματικής ρίζας (γραμμή 39)

Στη γραμμή «39» υπολογίζεται με την αριθμητική παράσταση «- b / (2 * a)» η διπλή πραγματική ρίζα της δευτεροβάθμιας αόριστης εξίσωσης και καταχωρείται στη μεταβλητή «x1_2».

Εκτύπωση της διπλής πραγματικής ρίζας (γραμμή 42)

Στη γραμμή «42» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «x1_2» (η) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

(+) Διακρίνουσα θετική (γραμμές 45-54)

Στις γραμμές «45-54» εκτελούνται οι εντολές της «else» (γραμμή 45) που αντιστοιχεί στην «if» της γραμμής «37» για την περίπτωση που η παράσταση «D == 0» παράγει αποτέλεσμα μία τιμή «False» (η διακρίνουσα «D» δεν περιέχει τιμή ίση με το «0» και είναι μεγαλύτερη του «0»).

Υπολογισμός της τετραγωνικής ρίζας της διακρίνουσας (γραμμή 47)

Στη γραμμή «47» υπολογίζεται με την αριθμητική παράσταση «sqrt(D)», η πραγματική τετραγωνική ρίζα διπλής ακριβείας της διακρίνουσας και το αποτέλεσμα καταχωρείται στη μεταβλητή «root_D».

Υπολογισμός της πρώτης πραγματικής ρίζας (γραμμή 48)

Στη γραμμή «48» υπολογίζεται με την αριθμητική παράσταση «(-b + root_D) / (2 * a)», η πρώτη πραγματική ρίζα διπλής ακριβείας της δευτεροβάθμιας εξίσωσης και το αποτέλεσμα καταχωρείται στη μεταβλητή «x1».

Υπολογισμός της δεύτερης πραγματικής ρίζας (γραμμή 49)

Στη γραμμή «49» υπολογίζεται με την αριθμητική παράσταση «(-b - root_D) / (2 * a)», η δεύτερη πραγματική ρίζα διπλής ακριβείας της δευτεροβάθμιας εξίσωσης και το αποτέλεσμα καταχωρείται στη μεταβλητή «x2».

Εκτύπωση της πρώτης πραγματικής ρίζας (γραμμή 51)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Στη γραμμή «51» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «x1» (η πρώτη πραγματική ρίζα της δευτεροβάθμιας εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση της δεύτερης πραγματικής ρίζας (γραμμή 52)

Στη γραμμή «52» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «x2» (η δεύτερη πραγματική ρίζα της δευτεροβάθμιας εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

(!) Αρνητική διακρίνουσα (γραμμές 56-61)

Στις γραμμές «56-61» εκτελούνται οι εντολές της «else» (γραμμή 56) για την περίπτωση που η παράσταση στην «if» ($D \geq 0$) της γραμμής «35» παράγει αποτέλεσμα μία τιμή «False» (η διακρίνουσα «D» δεν περιέχει τιμή ίση με το «0» και είναι μικρότερη του «0»). Στις γραμμές «58-59» τυπώνονται από τη «standard» έξοδο με την βοήθεια της συνάρτησης «printf()» μηνύματα για το είδος της εξίσωσης (αόριστη) και το πλήθος των πραγματικών ριζών της (0).

(~) Πρωτοβάθμια εξίσωση (γραμμές 63-70)

Στις γραμμές «63-70» εκτελούνται οι εντολές της «else» (γραμμή 63) που αντιστοιχεί στην «if» της γραμμής «30» για την περίπτωση που η παράσταση «a != 0» παράγει αποτέλεσμα μία τιμή «False» (ο συντελεστής «a» περιέχει την τιμή «0»).

Υπολογισμός της πραγματικής ρίζας (γραμμή 65)

Στη γραμμή «65» υπολογίζεται με την αριθμητική παράσταση «-c / b», η πραγματική ρίζα της πρωτοβάθμιας εξίσωσης και το αποτέλεσμα καταχωρείται στη μεταβλητή «x».

Εκτύπωση της πραγματικής ρίζας (γραμμή 68)

Στη γραμμή «68» τυπώνεται από τη «standard» έξοδο το περιεχόμενο

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

της μεταβλητής «x» (η πραγματική ρίζα της πρωτοβάθμιας εξίσωσης) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6lf», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1 (a == 0)

=====

Επίλυση δευτεροβάθμιας εξίσωσης

=====

Εισάγετε τον συντελεστή a : 0.0

Εισάγετε τον συντελεστή b : -11.0

Εισάγετε τον συντελεστή c : 3.0

Συντελεστής a : [0.000000]

Συντελεστής b : [-11.000000]

Συντελεστής c : [3.000000]

Η εξίσωση είναι πρωτοβάθμια

Πλήθος πραγματικών ριζών : 1

Ρίζα x : [0.272727]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Παράδειγμα 2 ($a \neq 0 / D \geq 0 / D == 0$)

=====

Επίλυση δευτεροβάθμιας εξίσωσης

=====

Εισάγετε τον συντελεστή a : 2.0

Εισάγετε τον συντελεστή b : 4.0

Εισάγετε τον συντελεστή c : 2.0

Συντελεστής a : [2.000000]

Συντελεστής b : [4.000000]

Συντελεστής c : [2.000000]

Η εξίσωση είναι δευτεροβάθμια

Διακρίνουσα : [0.000000]

Η εξίσωση είναι αόριστη

Η πραγματική ρίζα x1_2 είναι διπλή

Ρίζα x1_2 : [-1.000000]

Παράδειγμα 3 ($a \neq 0 / D \geq 0 / D \neq 0$)

=====

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Επίλυση δευτεροβάθμιας εξίσωσης

=====

Εισάγετε τον συντελεστή a : 2.8

Εισάγετε τον συντελεστή b : -7.1

Εισάγετε τον συντελεστή c : -1.6

Συντελεστής a : [2.800000]

Συντελεστής b : [-7.100000]

Συντελεστής c : [-1.600000]

Η εξίσωση είναι δευτεροβάθμια

Διακρίνουσα : [68.330000]

Πλήθος πραγματικών ριζών : 2

Ρίζα x1 : [2.743964]

Ρίζα x2 : [-0.208249]

Παράδειγμα 4 (a != 0 / D < 0)

=====

Επίλυση δευτεροβάθμιας εξίσωσης

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

=====

Εισάγετε τον συντελεστή a : 32.679

Εισάγετε τον συντελεστή b : 1.0

Εισάγετε τον συντελεστή c : 16.21456

Συντελεστής a :	[32.679000]
Συντελεστής b :	[1.000000]
Συντελεστής c :	[16.214560]

Η εξίσωση είναι δευτεροβάθμια

Διακρίνουσα : [-2118.502425]

Η εξίσωση είναι αδύνατη

Πλήθος πραγματικών ριζών : 0

ΠΑΡΑΤΗΡΗΣΕΙΣ

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)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Στο «Παράδειγμα 1» διαβάζεται από τη «standard» είσοδο ο πρώτος συντελεστής «a» της εξίσωσης (0), ο δεύτερος συντελεστής «b» (-11) και ο τρίτος «c» (3). Ορθά τυπώνονται οι τρεις αυτοί αριθμοί, πραγματοποιείται ο πρώτος έλεγχος για το περιεχόμενο του «a» που περιέχει τη τιμή «0», οπότε πρόκειται για πρωτοβάθμια εξίσωση με μία πραγματική ρίζα και ορθά τυπώνεται η ρίζα αυτή (x). Τα αποτελέσματα :

a : 0.000000

b : -11.000000

c : 3.000000

x : 0.272727

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

Στο «Παράδειγμα 2» διαβάζεται από τη «standard» είσοδο ο πρώτος συντελεστής «a» της εξίσωσης (2), ο δεύτερος συντελεστής «b» (4) και ο τρίτος «c» (4). Ορθά τυπώνονται οι τρεις αυτοί αριθμοί, πραγματοποιείται ο πρώτος έλεγχος για το περιεχόμενο του «a» που δεν περιέχει την τιμή «0», οπότε πρόκειται για δευτεροβάθμια εξίσωση. Ορθά υπολογίζεται και τυπώνεται το αποτέλεσμα της διακρίνουσας (D), πραγματοποιείται ο δεύτερος έλεγχος για το περιεχόμενο της «D» που περιέχει τιμή μεγαλύτερη ή ίση του «0», οπότε έχει σίγουρα πραγματικές ρίζες. Πραγματοποιείται ο τρίτος έλεγχος για το περιεχόμενο της «D» που περιέχει την τιμή «0», οπότε πρόκειται για αόριστη εξίσωση με διπλή πραγματική ρίζα (x1_2). Τα αποτελέσματα :

a : 2.000000

b : 4.000000

c : 2.000000

D : 0.000000

x1_2 : -1.000000

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

Στο «Παράδειγμα 3» διαβάζεται από τη «standard» είσοδο ο πρώτος συντελεστής «a» της εξίσωσης (2.8), ο δεύτερος συντελεστής «b» (-7.1)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

και ο τρίτος «c» (-1.6). Ορθά τυπώνονται οι τρεις αυτοί αριθμοί, πραγματοποιείται ο πρώτος έλεγχος για το περιεχόμενο του «a» που δεν περιέχει την τιμή «0», οπότε πρόκειται για δευτεροβάθμια εξίσωση. Ορθά υπολογίζεται και τυπώνεται το αποτέλεσμα της διακρίνουσας (68.33), πραγματοποιείται ο δεύτερος έλεγχος για το περιεχόμενο της «D» που περιέχει τιμή μεγαλύτερη ή ίση του «0», οπότε έχει σίγουρα πραγματικές ρίζες. Πραγματοποιείται ο τρίτος έλεγχος για το περιεχόμενο της «D» που δεν περιέχει την τιμή «0», οπότε πρόκειται για εξίσωση με δύο πραγματικές ρίζες (x_1 , x_2). Τα αποτελέσματα :

a : 2.800000

b : -7.100000

c : -1.600000

D : 68.300000

x_1 : 2.743964

x_2 : -0.208249

Παράδειγμα 4 ($a \neq 0 / D < 0$) :

Στο «Παράδειγμα 4» διαβάζεται από τη «standard» είσοδο ο πρώτος συντελεστής «a» της εξίσωσης (32.679), ο δεύτερος συντελεστής «b» (1) και ο τρίτος «c» 16.21456). Ορθά τυπώνονται οι τρεις αυτοί αριθμοί, πραγματοποιείται ο πρώτος έλεγχος για το περιεχόμενο του «a» που δεν περιέχει την τιμή «0», οπότε πρόκειται για δευτεροβάθμια εξίσωση. Ορθά υπολογίζεται και τυπώνεται το αποτέλεσμα της διακρίνουσας (-2118.502425), πραγματοποιείται ο δεύτερος έλεγχος για το περιεχόμενο της «D» που περιέχει τιμή μικρότερη του «0», οπότε πρόκειται για αδύνατη εξίσωση χωρίς πραγματικές ρίζες. Τα αποτελέσματα :

a : 32.679000

b : 1.000000

c : 16.214560

D : -2118.502425

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΘΕΜΑ 4

ΕΠΙΣΗΜΑΝΣΗ «MaxInteger.c»

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

ΠΡΟΓΡΑΜΜΑ «MaxInteger.c»

```
1  #include <stdio.h>
2
3  int main (int argc, int **argv)
4
5  {
6
7      system ("chcp 1253");
8
9      int a, b, c; // Δήλωση μεταβλητών
10
11     printf ("=====\n\n");
12     printf ("Εύρεση μέγιστου αριθμού και σειράς που εισήχθη\n\n");
13     printf ("=====\n\n");
14     printf ("Εισάγετε τον 1ο ακέραιο αριθμό : ");
15     scanf ("%d", &a); // Εισαγωγή του πρώτου ακεραίου αριθμού
16     printf ("Εισάγετε τον 2ο ακέραιο αριθμό : ");
17     scanf ("%d", &b); // Εισαγωγή του δεύτερου ακεραίου αριθμού
18     printf ("Εισάγετε τον 3ο ακέραιο αριθμό : ");
19     scanf ("%d", &c); // Εισαγωγή του τρίτου ακεραίου αριθμού
20     printf ("\n-----\n\n");
21     printf ("Ο 1ος ακέραιος αριθμός : [%20d]\n", a);
22     printf ("Ο 2ος ακέραιος αριθμός : [%20d]\n", b);
23     printf ("Ο 3ος ακέραιος αριθμός : [%20d]\n\n", c);
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
24      (a > b && a > c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 1ος.", a) : printf ("\b"); // a > b ΚΑΙ a > c

25      (a < b && b > c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 2ος.", b) : printf ("\b"); // a < b ΚΑΙ b > c

26      (a < c && b < c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 3ος.", c) : printf ("\b"); // a < c ΚΑΙ b < c

27      (a == b && a > c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 1ος και 2ος.", a) : printf ("\b"); // a == b > c

28      (a < b && b == c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 2ος και 3ος.", b) : printf ("\b"); // a < b == c

29      (a > b && a == c) ? printf ("Ο μεγαλύτερος αριθμός είναι ο αριθμός
%d και εισήχθη 1ος και 3ος.", c) : printf ("\b"); // a == c > b

30      (a == b && b == c && a == c) ? printf ("Ο μεγαλύτερος αριθμός
είναι ο αριθμός %d και εισήχθη 1ος, 2ος και 3ος.\n", a) : printf ("\n");
// a == b == c

31

33      return 0;

33

34 }
```

ΤΕΚΜΗΡΙΩΣΗ «MaxInteger.c»

ΖΗΤΟΥΜΕΝΟ

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

- a) Διαβάζει από τη «standard» είσοδο τρεις αριθμούς στο σύνολο των ακεραίων (τύπου int).
- b) Συγκρίνει τους τρεις αριθμούς, οι οποίοι έχουν καταχωρηθεί στα περιεχόμενα τριών μεταβλητών.
- c) Τυπώνει τα κατάλληλα μηνύματα που διασαφηνίζουν 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.

ΔΟΜΗ

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

- a) «stdio.h» : Περιέχει τις έτοιμες συναρτήσεις «scanf()» και «printf()» που συνδέονται με τα κανάλια εισόδου και εξόδου αντίστοιχα για την αναγνώ-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ση και την τύπωση 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 :

a) συνθήκης : {? :}

b) σχεσιακοί : <, >, ==

c) ανάθεσης : =

d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης της «standard» εισόδου, «scanf()».

ΜΕΤΑΒΛΗΤΕΣ

Ακέραιες μεταβλητές (τύπου int)

a (ο πρώτος ακέραιος αριθμός)

b (ο δεύτερος ακέραιος αριθμός)

c (ο τρίτος ακέραιος αριθμός)

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών (γραμμή 9)

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στην γραμμή «9» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Τίτλος προγράμματος (γραμμή 12)

Στην γραμμή «12» 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).

Εισαγωγή του πρώτου ακεραίου αριθμού (γραμμή 15)

Στη γραμμή «15» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο πρώτος ακέραιος αριθμός. Στη συνέχεια, με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

μήνυμα (γραμμή 14).

Εισαγωγή του δεύτερου ακεραίου αριθμού **(γραμμή 17)**

Στη γραμμή «17» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο δεύτερος ακεραίος αριθμός. Στη συνέχεια, με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 16).

Εισαγωγή του τρίτου ακεραίου αριθμού **(γραμμή 19)**

Στη γραμμή «19» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο ο τρίτος ακεραίος αριθμός. Στη συνέχεια, με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 18).

Εκτύπωση του πρώτου ακεραίου αριθμού **(γραμμή 21)**

Στη γραμμή «21» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «a» (ο πρώτος ακεραίος αριθμός) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του δεύτερου ακεραίου αριθμού **(γραμμή 22)**

Στη γραμμή «22» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «b» (ο δεύτερος ακεραίος αριθμός) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του τρίτου ακεραίου αριθμού **(γραμμή 23)**

Στη γραμμή «23» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «c» (ο τρίτος ακεραίος αριθμός) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

a > b ΚΑΙ a > c **(γραμμή 24)**

Στη γραμμή «24» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

αριθμός καταχωρείται στη μεταβλητή «x» και εισήχθη πρώτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

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

$a < b \text{ KAI } b > c$ (γραμμή 25)

Στη γραμμή «25» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στη μεταβλητή «b» και εισήχθη δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

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

$a < c \text{ KAI } b < c$ (γραμμή 26)

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

a == b > c

(γραμμή 27)

Στην γραμμή «27» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «a» και «b» και εισήχθη πρώτος και δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

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

a < b == c

(γραμμή 28)

Στην γραμμή «28» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «b» και «c» και εισήχθη δεύτερος και τρίτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

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

a == c > b

(γραμμή 29)

Στην γραμμή «29» συντάσσεται η περίπτωση, όπου ο μεγαλύτερος αριθμός καταχωρείται στις μεταβλητές «a» και «c» και εισήχθη πρώτος και δεύτερος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

Εφόσον, η παράσταση «a > b && a == c» παράγει αποτέλεσμα μία τιμή «True» (τιμή διάφορη του «0»), δηλαδή, οι μεταβλητές «a» και «c»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

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

a == b == c

(γραμμή 30)

Στην γραμμή «30» συντάσσεται η περίπτωση, όπου ο ίδιος αριθμός καταχωρείται και στις τρεις μεταβλητές «a», «b», «c» και εισήχθη πρώτος, δεύτερος και τρίτος. Η χρήση του τριαδικού τελεστή συνθήκης «? :» δικαιολογείται ως εξής :

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

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1 (a > b ΚΑΙ a > c / 1^{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

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο 1ος ακέραιος αριθμός : [8]

Ο 2ος ακέραιος αριθμός : [-2]

Ο 3ος ακέραιος αριθμός : [0]

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

Παράδειγμα 2 ($a < b$ ΚΑΙ $b > c$ / 2^{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

=====

Ο 1ος ακέραιος αριθμός : [2]

Ο 2ος ακέραιος αριθμός : [5]

Ο 3ος ακέραιος αριθμός : [4]

Ο μεγαλύτερος αριθμός είναι ο αριθμός 5 και εισήχθη 2ος.

Παράδειγμα 3 ($a < c$ ΚΑΙ $b < c$ / 3^{ος})

=====

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

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 1ος ακέραιος αριθμός : [-15]

Ο 2ος ακέραιος αριθμός : [-13]

Ο 3ος ακέραιος αριθμός : [-9]

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

Παράδειγμα 4 ($a == b > c / 1^{ος}, 2^{ος}$)

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

Εισάγετε τον 1ο ακέραιο αριθμό : 32

Εισάγετε τον 2ο ακέραιο αριθμό : 32

Εισάγετε τον 3ο ακέραιο αριθμό : 1

Ο 1ος ακέραιος αριθμός : [32]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο 2ος ακέραιος αριθμός : [32]

Ο 3ος ακέραιος αριθμός : [1]

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

Παράδειγμα 5 ($a < b == c / 2^{\text{ος}}, 3^{\text{ος}}$)

=====

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 1ος ακέραιος αριθμός : [-20]

Ο 2ος ακέραιος αριθμός : [-4]

Ο 3ος ακέραιος αριθμός : [-4]

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

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

=====

Εύρεση μέγιστου αριθμού και σειράς που εισήχθη

=====

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εισάγετε τον 1ο ακέραιο αριθμό : 18

Εισάγετε τον 2ο ακέραιο αριθμό : 4

Εισάγετε τον 3ο ακέραιο αριθμό : 18

Ο 1ος ακέραιος αριθμός : [18]

Ο 2ος ακέραιος αριθμός : [4]

Ο 3ος ακέραιος αριθμός : [18]

Ο μεγαλύτερος αριθμός είναι ο αριθμός 18 και εισήχθη 1ος και 3ος.

Παράδειγμα 7 (a == b == c / 1^{ος} , 2^{ος} , 3^{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 1ος ακέραιος αριθμός : [100]

Ο 2ος ακέραιος αριθμός : [100]

Ο 3ος ακέραιος αριθμός : [100]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

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

ΠΑΡΑΤΗΡΗΣΕΙΣ

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

Παράδειγμα 1 ($a > b$ ΚΑΙ $a > c$ / 1^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «a» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «b» και «c» και εισήχθη πρώτος. Τα αποτελέσματα :

a : 8

b : -2

c : 0

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

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

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «b» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «a» και «c» και εισήχθη δεύτερος. Τα αποτελέσματα :

a : 2

b : 5

c : 4

Ο μεγαλύτερος αριθμός είναι ο αριθμός 5 και εισήχθη 2ος.

Παράδειγμα 3 ($a < c$ ΚΑΙ $b < c$ / 3^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «c» περιέχει αριθμό μεγαλύτερο από τους αντίστοιχους των «a» και «b» και εισήχθη τρίτος. Τα αποτελέσματα :

a : -15

b : -13

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

c : -9

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

Παράδειγμα 4 ($a == b > c$ / 1^{ος}, 2^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου η μεταβλητή «a» και «b» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «c» και εισήχθη πρώτος και δεύτερος. Τα αποτελέσματα :

a : 32

b : 32

c : 1

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

Παράδειγμα 5 ($a < b == c$ / 2^{ος}, 3^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «b» και «c» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «a» και εισήχθη δεύτερος και τρίτος. Τα αποτελέσματα :

a : -20

b : -20

c : -4

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

Παράδειγμα 6 ($a == c > b$ / 1^{ος}, 3^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «a» και «c» περιέχουν τον ίδιο αριθμό που είναι μεγαλύτερος από τον αντίστοιχο της «b» και εισήχθη πρώτος και τρίτος. Τα αποτελέσματα :

a : 18

b : 4

c : 18

Ο μεγαλύτερος αριθμός είναι ο αριθμός 18 και εισήχθη 1ος και 3ος.

Παράδειγμα 7 ($a == b == c$ / 1^{ος}, 2^{ος}, 3^{ος}) :

Αντιπροσωπεύει την περίπτωση όπου οι μεταβλητές «a», «b» και «c»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

περιέχουν τον ίδιο και εισήχθη πρώτος, δεύτερος και τρίτος. Τα αποτελέσματα :

a : 100

b : 100

c : 100

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



Σας ευχαριστώ για την προσοχή σας.

