

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΔΥΤΙΚΗΣ ΑΤΤΙΚΗΣ
UNIVERSITY OF WEST ATTICA

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ
ΥΠΟΛΟΓΙΣΤΩΝ

ΕΡΓΑΣΙΑ 4 ΒΡΟΧΟΙ

ΣΤΟΙΧΕΙΑ ΦΟΙΤΗΤΗ:

ΟΝΟΜΑΤΕΠΩΝΥΜΟ: ΑΘΑΝΑΣΙΟΥ ΒΑΣΙΛΕΙΟΣ ΕΥΑΓΓΕΛΟΣ

ΑΡΙΘΜΟΣ ΜΗΤΡΩΟΥ: 19390005

ΚΑΤΑΣΤΑΣΗ ΦΟΙΤΗΤΗ: ΠΡΟΠΤΥΧΙΑΚΟ

ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ: ΠΑΔΑ

ΤΜΗΜΑ ΕΡΓΑΣΤΗΡΙΟΥ: Μ2

ΚΑΘΗΓΗΤΗΣ ΕΡΓΑΣΤΗΡΙΟΥ: ΓΕΩΡΓΙΟΣ ΜΕΛΕΤΙΟΥ

ΗΜΕΡΟΜΗΝΙΑ ΟΛΟΚΛΗΡΩΣΗΣ ΤΗΣ ΕΡΓΑΣΙΑΣ: 7/12/2021

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΕΡΙΕΧΟΜΕΝΑ

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ (ΣΕΛΙΔΕΣ 4-9)

ΘΕΜΑ 1	(ΣΕΛΙΔΕΣ 4-6)
ΘΕΜΑ 1a) Τι είναι οι επαναληπτικές δομές (βρόχοι);	(ΣΕΛΙΔΑ 4)
ΘΕΜΑ 1b) Ποιες επαναληπτικές δομές γνωρίζετε στη «C»;	(ΣΕΛΙΔΕΣ 4-6)
ΘΕΜΑ 1c) Πώς σχετίζονται μεταξύ τους;	(ΣΕΛΙΔΑ 6)
ΘΕΜΑ 2	(ΣΕΛΙΔΕΣ 7-9)
ΘΕΜΑ 2a) Τι γνωρίζετε για την δομή επανάληψης «for»;	(ΣΕΛΙΔΕΣ 7-8)
ΘΕΜΑ 2b) Τι γνωρίζετε για την εντολή «break»;	(ΣΕΛΙΔΑ 8)
ΘΕΜΑ 2c) Τι γνωρίζετε για την εντολή «continue»;	(ΣΕΛΙΔΕΣ 8-9)

ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ (ΣΕΛΙΔΕΣ 9-77)

ΘΕΜΑ 3 (ΣΕΛΙΔΕΣ 9-39)

ΘΕΜΑ 3Α	(ΣΕΛΙΔΕΣ 9-22)
Επισήμανση «MathsWithIntegersInLoopA.c»	(ΣΕΛΙΔΑ 9)
Πρόγραμμα «MathsWithIntegersInLoopA.c»	(ΣΕΛΙΔΕΣ 9-11)
Τεκμηρίωση «MathsWithIntegersInLoopA.c»	(ΣΕΛΙΔΕΣ 11-24)
{	
Ζητούμενο	(ΣΕΛΙΔΕΣ 11-12)
Δομή	(ΣΕΛΙΔΕΣ 12-13)
Μεταβλητές	(ΣΕΛΙΔΑ 13)
Διάσχιση	(ΣΕΛΙΔΕΣ 13-21)
Παραδείγματα	(ΣΕΛΙΔΕΣ 21-23)
Παρατηρήσεις	(ΣΕΛΙΔΕΣ 23-24)
}	

ΘΕΜΑ 3Β	(ΣΕΛΙΔΕΣ 24-39)
Επισήμανση «MathsWithIntegersInLoopB.c»	(ΣΕΛΙΔΑ 24)
Πρόγραμμα «MathsWithIntegersInLoopB.c»	(ΣΕΛΙΔΕΣ 24-27)
Τεκμηρίωση «MathsWithIntegersInLoopB.c»	(ΣΕΛΙΔΕΣ 27-39)
{	
Ζητούμενο	(ΣΕΛΙΔΑ 27)
Δομή	(ΣΕΛΙΔΑ 27-28)
Μεταβλητές	(ΣΕΛΙΔΑ 28)
Διάσχιση	(ΣΕΛΙΔΕΣ 29-35)
Παραδείγματα	(ΣΕΛΙΔΕΣ 35-38)
Παρατηρήσεις	(ΣΕΛΙΔΑ 39)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

}

ΘΕΜΑ 4 (ΣΕΛΙΔΕΣ 39-62)

Επισήμανση «Stars.c» (ΣΕΛΙΔΕΣ 39-40)

Πρόγραμμα «Stars.c» (ΣΕΛΙΔΕΣ 40-42)

Τεκμηρίωση «Stars.c» (ΣΕΛΙΔΕΣ 42-62)

{

 Ζητούμενο (ΣΕΛΙΔΕΣ 42-43)

 Δομή (ΣΕΛΙΔΕΣ 43-44)

 Μεταβλητές (ΣΕΛΙΔΑ 44)

 Διάσχιση (ΣΕΛΙΔΕΣ 44-59)

 Παραδείγματα (ΣΕΛΙΔΕΣ 59-62)

 Παρατηρήσεις (ΣΕΛΙΔΑ 62)

}

ΘΕΜΑ 5 (ΣΕΛΙΔΕΣ 63-77)

Επισήμανση «SinTaylor.c» (ΣΕΛΙΔΑ 63)

Πρόγραμμα «SinTaylor.c» (ΣΕΛΙΔΕΣ 63-65)

Τεκμηρίωση «SinTaylor.c» (ΣΕΛΙΔΕΣ 65-77)

{

 Ζητούμενο (ΣΕΛΙΔΕΣ 65-66)

 Δομή (ΣΕΛΙΔΕΣ 66-67)

 Μεταβλητές (ΣΕΛΙΔΑ 67)

 Διάσχιση (ΣΕΛΙΔΕΣ 68-73)

 Παραδείγματα (ΣΕΛΙΔΕΣ 73-76)

 Παρατηρήσεις (ΣΕΛΙΔΕΣ 76-77)

}

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΑΠΑΝΤΗΣΕΙΣ ΣΤΑ ΘΕΜΑΤΑ ΘΕΩΡΙΑΣ

ΘΕΜΑ 1

a) Τι είναι οι επαναληπτικές δομές (βρόχοι);

Οι επαναληπτικές δομές είναι οι εντολές που εκτελούν ένα σώμα εντολών επανηλειμμένα για συγκεκριμένο πλήθος φορών. Το πλήθος των φορών καθορίζεται από την τιμή μίας παράστασης που χρησιμοποιείται ως συνθήκη του βρόχου και συγκεκριμένα, για όσο η τιμή της παράστασης που παράγεται είναι διάφορη του «0» (τιμή «True»), τότε, ο βρόχος θα εκτελείται έως ότου η τιμή της παράστασης που παράγεται είναι «0» (τιμή «False»).

b) Ποιες επαναληπτικές δομές γνωρίζετε στη γλώσσα «C»;

Οι επαναληπτικές δομές της γλώσσας προγραμματισμού «C» είναι οι εξής :

a) while :

Λειτουργία

Εκτελεί ένα σώμα εντολών για όσες φορές η τιμή της παράστασης παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»).

Σύνταξη

```
while (παράσταση)
{
```

Εντολές

```
}
```

Παράδειγμα :

```
x = -5;

while (x < 0)
{
    printf ("Αρνητικός αριθμός");
    x++;
}
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Επεξήγηση παραδείγματος

Αρχικά, καταχωρείται στη μεταβλητή «x» η τιμή «-5». Η εντολή «while» εκτελεί το σώμα εντολών, για όσο η μεταβλητή «x» περιέχει τιμή μικρότερη του «0», δηλαδή, για όσο η παράσταση «x < 0» παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»). Το σώμα εντολών περιέχει μία συνάρτηση «printf()» που τυπώνει το μήνυμα «Αρνητικός αριθμός» για να δηλωθεί ότι η «x» περιέχει αρνητική τιμή και την αριθμητική πράξη «x++» που αυξάνει το περιεχόμενο της μεταβλητής «x» κατά «1». Επομένως, ο βρόχος θα εκτελεστεί «5» φορές για όσο η μεταβλητή «x» περιέχει τις τιμές «-5», «-4», «-3», «-2» και «-1».

b) do – while :

Λειτουργία

Εκτελεί ένα σώμα εντολών, τουλάχιστον, μία φορά και για όσες φορές η τιμή της παράστασης παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»).

Σύνταξη

```
do  
{
```

Εντολές

```
}
```

```
while (παράσταση);
```

Παράδειγμα

```
x = 5;  
  
do  
{  
    printf ("Αριθμός : 5");  
}  
  
while (x != 5);
```

Επεξήγηση παραδείγματος

Αρχικά, καταχωρείται στη μεταβλητή «x» η τιμή «5». Η εντολή «do – while» θα εκτελέσει μία φορά το σώμα εντολών και μετά για όσο η μεταβλητή «x» δεν περιέχει την τιμή «5», δηλαδή, για όσο η παράσταση «x != 5» παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»). Το σώμα εντολών περιέχει μία συνάρτηση «printf()» που τυπώνει το μήνυμα «Αριθμός : 5» για να δηλωθεί ότι η «x» περιέχει την τιμή «5». Επομένως, ο βρόχος θα εκτελεστεί «1» φορά

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ανεξαρτήτως του περιεχομένου της μεταβλητής «x».

c) for :

Λειτουργία

Εκτελεί ένα σώμα εντολών για όσες φορές η τιμή της παράστασης παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»).

Σύνταξη

```
for (παράσταση1 ; παράσταση2 ; παράσταση3)  
{
```

Εντολές

```
}
```

Παράδειγμα

```
for (i = 0 ; i < 3 ; i++)  
{  
    printf ("Hello world");  
}
```

Επεξήγηση παραδείγματος

Αρχικά, καταχωρείται στη μεταβλητή «i» η τιμή «0». Η εντολή «for» θα εκτελέσει το σώμα εντολών για όσο η μεταβλητή «i» περιέχει τιμή μικρότερη του «3», δηλαδή, για όσο η παράσταση «i < 3» παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»). Το σώμα εντολών περιέχει μία συνάρτηση «printf()» που τυπώνει το μήνυμα «Hello world». Στη συνέχεια, εκτελείται και η παράσταση «i++» που αυξάνει την τιμή της μεταβλητής «i» κατά «1». Επομένως, ο βρόχος θα εκτελεστεί «3» φορές για όσο η μεταβλητή «i» περιέχει τις τιμές «0», «1» και «2».

c) Πώς σχετίζονται μεταξύ τους:

Οι επαναληπτικές δομές «while», «do – while» και «for» έχουν κοινή λειτουργία, να εκτελούν ένα σώμα εντολών επανειλημμένα για συγκεκριμένο πλήθος φορών. Πιο αναλυτικά, ο βρόχος εκτελείται για όσο η τιμή της παράστασης, που ελέγχουν οι εντολές επανάληψης για να καθοριστεί το πλήθος φορών του βρόχου, παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0») με την διαφορά να κάνει η «do – while» που πρώτα θα εκτελέσει μία φορά το σώμα εντολών και μετά θα ελέγξει την τιμή της παράστασης. Ειδαλλώς, ο βρόχος διακόπτεται.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΘΕΜΑ 2

a) Τι γνωρίζετε για την δομή επανάληψης «for»;

Η δομή επανάληψης «for» εκτελεί ένα σώμα εντολών επανειλημμένα για συγκεκριμένο πλήθος φορών (βλ. Θέμα 1b, σελίδα «6», «c) for») και συντάσσεται ως εξής :

```
for (παράσταση1 ; παράσταση2 ; παράσταση3)
{
    Εντολές
}
```

Πιο αναλυτικά, ο βρόχος εκτελείται για όσο η «παράσταση2» παράγει αποτέλεσμα μία τιμή «True» (τιμή διάφορη του «0»). Η «παράσταση2», συνήθως, είναι μία παράσταση με λογικούς ή σχεσιακούς τελεστές, ώστε κατά την εκτέλεση της εντολής η τιμή που μπορεί να παράγει η «παράσταση2» να είναι η τιμή «False» (τιμή «0»), ειδαλλώς ο βρόχος θα εκτελείται και δεν θα σταματήσει ποτέ (ατέρμων βρόχος). Η «παράσταση1» συνήθως είναι μια αρχική καταχώρηση βοηθητικής μεταβλητής (μετρητής) που χρησιμοποιείται για να ελέγχει ο χρήστης τον βρόχο που εκτελείται, δηλαδή, να ορίζει το πλήθος των φορών εκτέλεσης του βρόχου. Τέλος, η «παράσταση3» είναι μία αριθμητική παράσταση που τροποποιεί το περιεχόμενο της βοηθητικής μεταβλητής που ορίστηκε ως «μετρητής» στη θέση «παράσταση1», προκειμένου να επιτυγχάνεται η ομαλή λειτουργία του βρόχου και να παράγεται το επιθυμητό αποτέλεσμα. Οι τρεις παραστάσεις χωρίζονται με τον χαρακτήρα του ελληνικού ερωτηματικού «;».

Η «for» είναι για τους προγραμματιστές η πιο «βολική» εντολή (όχι σε όλες τις περιπτώσεις) επανάληψης για τους λόγους ότι δέχεται τρία ορίσματα με τρεις ή και περισσότερες παραστάσεις εξοικονομώντας χώρο και κάνοντας τον κώδικα πιο ευανάγνωστο.

Τα παρακάτω παραδείγματα εκτελούν επανειλημμένα μία συνάρτηση «printf()» που τυπώνει το μήνυμα «Θετικός» που δηλώνει ότι το περιεχόμενο της μεταβλητής «x» είναι θετικός αριθμός. Το πρώτο παράδειγμα υλοποιείται με την εντολή επανάληψης «while» , ενώ το δεύτερο με την «for». Η τιμή της μεταβλητής «x» αλλάζει με την παράσταση «x- -», οπότε ο βρόχος εκτελείται «1» φορά για «x = 1».

Παράδειγμα 1

```
x = 1
while (x > 0)
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
{  
    printf ("Θετικός");  
    x--  
}
```

Παράδειγμα 2

```
for (x = 1 ; x > 0 ; x--)  
{  
    printf ("Θετικός");  
}
```

b) Τι γνωρίζετε για την εντολή «break»:

Η εντολή «break» διακόπτει την πορεία του προγράμματος από την γραμμή που βρίσκεται έως το τέλος μίας γραμμής εντολών. Πιο συγκεκριμένα, σ' έναν βρόχο η «break» ουσιαστικά ακυρώνει τον βρόχο στο σημείο όπου βρίσκεται, ανεξάρτητα τι τιμή παράγει η παράσταση που τον ελέγχει μέσα σ' αυτόν και μεταφέρει το πρόγραμμα στις εντολές που βρίσκονται έξω απ' αυτόν.

Για παράδειγμα :

```
for (x = 0 ; x < 100 ; x++)  
{  
    printf ("Μικρότερος του 100"); break;  
    printf ("Θετικός");  
}  
printf ("Τιμή 0!");
```

Στο παράδειγμα η «for» για «x = 0» ελέγχει και βρίσκει ότι η παράσταση «x < 100» παράγει αποτέλεσμα μία τιμή «True» (διάφορη του 0), επομένως θα εκτελεστεί ο βρόχος. Η «printf()» μέσα στον βρόχο τυπώνει το μήνυμα «Μικρότερος του 100» από τη «standard» έξοδο και η «break» ακυρώνει τον βρόχο ο οποίος θα συνεχιζόταν μέχρι η μεταβλητή «x» περιείχε την τιμή «100» και το πρόγραμμα πηγαίνει στην εντολή «printf()» που βρίσκεται έξω από τον βρόχο και τυπώνεται το μήνυμα «Τιμή 0!». Συνοψίζοντας, ο βρόχος θα εκτελεστεί «1» φορά μόνο και η εντολή «printf ("Θετικός");» δεν θα εκτελεστεί ποτέ.

c) Τι γνωρίζετε για την εντολή «continue»:

Η εντολή «continue» διακόπτει την πορεία του βρόχου από το σημείο που αυτή βρίσκεται και επαναφέρει το πρόγραμμα από την αρχή του βρόχου.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Πιο αναλυτικά, δεν εκτελούνται οι εντολές του βρόχου που βρίσκονται κάτω από την «continue», καθώς το πρόγραμμα μεταφέρεται στην αρχή του βρόχου και εκτελείται από την αρχή έως την γραμμή που βρίσκεται η εντολή αυτή όσες φορές η παράσταση που εκφράζει τον βρόχο παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»).

Για παράδειγμα :

```
for (x = -3 ; x < 0 ; x++)
{
    printf ("Αρνητικός"); continue;
    printf ("Θετικός");
}
printf ("Τιμή 0!");
```

Στο παράδειγμα η «for» για «x = -3» ελέγχει και βρίσκει ότι η παράσταση «x < 0» παράγει αποτέλεσμα μία τιμή «True» (διάφορη του «0»), επομένως θα εκτελεστεί ο βρόχος. Η «printf()» μέσα στον βρόχο τυπώνει το μήνυμα «Αρνητικός» από τη «standard» έξοδο και η «continue» επαναφέρει τον βρόχο από την αρχή και εκτελείται στην συνέχεια για «x = -2» και για «x = -1». Η εντολή «printf ("Θετικός\n");» που βρίσκεται κάτω από την «continue» δεν θα εκτελεστεί ποτέ. Συνοψίζοντας, ο βρόχος θα εκτελεστεί «3» φορές και θα τυπώνει το μήνυμα «Αρνητικός» από την «printf()» του βρόχου και όταν τελειώσει θα εκτελεστεί η εντολή «printf ("Τιμή 0!");» που βρίσκεται έξω από τον βρόχο, όπου τυπώνει το μήνυμα «Τιμή 0!» δηλώνοντας ότι η μεταβλητή «x» περιέχει την τιμή «0».

ΠΗΓΑΙΟΙ ΚΩΔΙΚΕΣ / ΤΕΚΜΗΡΙΩΣΕΙΣ

ΘΕΜΑ 3Α

ΕΠΙΣΗΜΑΝΣΗ «MathsWithIntegersInLoopA.c»

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

Πιο αναλυτικά, στην περίπτωση όπου ο χρήστης εισάγει όσους ακραίους θέλει και ο βρόχος σταματάει μέχρι ο χρήστης εισάγει όλους τους αριθμούς που έχει δηλώσει ότι θα εισάγει.

ΠΡΟΓΡΑΜΜΑ «MathsWithIntegersInLoopA.c»

```
1 #include <stdio.h>
2
3 int main (int argc, char **argv)
4
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```

5 {
6     system ("chcp 1253");
7
8     int N, n; // Δήλωση μεταβλητών
9     int i = 0; // Αρχικοποίηση μεταβλητών
10    int pow;
11    int evens = 0; // Αρχικοποίηση μεταβλητών
12    int neg = 0; // Αρχικοποίηση μεταβλητών
13    int prod = 1; // Αρχικοποίηση μεταβλητών
14    int pos = 0; // Αρχικοποίηση μεταβλητών
15    int sum = 0; // Αρχικοποίηση μεταβλητών
16    float avg;
17
18    printf ("=====\n\n");
19    printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
20    // Τίτλος του προγράμματος
21    printf ("=====\n\n");
22    printf ("Εισάγετε N ακέραιους αριθμούς : ");
23    scanf ("%d", &N); // Εισαγωγή του πλήθους των ακεραίων αριθμών
24    printf ("\n-----\n\n");
25    printf ("Το πλήθος των ακεραίων : [%20d]\n\n", N);
26    // Εκτύπωση του πλήθους των ακεραίων αριθμών
27    if (N > 0) /* (~) Ένας ή περισσότεροι ακέραιοι αριθμοί */
28    {
29        while (i < N) /* Βρόχος */
30        {
31            printf ("-----\n\n");
32            printf ("Εισάγετε ακεραίο αριθμό : ");
33            scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών
34            printf ("\n-----\n\n");
35            printf ("Ο ακεραίος αριθμός : [%20d]\n\n", n);
36            // Εκτύπωση των ακεραίων αριθμών
37            if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
38            {
39                pow = n * n; // Υπολογισμός του τετραγώνου του
40                περιττού
41                printf ("Τετράγωνο περιττού : [%20d]\n\n",
42                pow); // Εκτύπωση του τετραγώνου του περιττού
43                printf ("Περιττός αριθμός\n");
44            }
45            else /* (!) Εισαγωγή άρτιου αριθμού */
46            {
47                evens++; // Υπολογισμός του πλήθους των άρτιων
48                αριθμών
49                printf ("\nΆρτιος αριθμός\n");
50            }
51            if (n >= 0) /* (+) Εισαγωγή θετικού αριθμού */
52            {
53                pos++; // Υπολογισμός του πλήθους των θετικών
54                αριθμών
55                sum = sum + n; // Υπολογισμός του αθροίσματος των
56                θετικών αριθμών
57                avg = (float) sum / pos; // Υπολογισμός του μέσου
58                όρου των θετικών αριθμών
59            }
60            else /* (+) Εισαγωγή αρνητικού αριθμού */
61            {
62                neg--; // Υπολογισμός του πλήθους των αρνητικών
63                αριθμών
64                sum = sum - n; // Υπολογισμός του αθροίσματος των
65                αρνητικών αριθμών
66                avg = (float) sum / neg; // Υπολογισμός του μέσου
67                όρου των αρνητικών αριθμών
68            }
69            i++;
70        }
71    }
72    printf ("\nΜέση τιμή : [%20f]\n\n", avg);
73    // Εκτύπωση του μέσου όρου
74    printf ("=====\n\n");
75    // Τίτλος του προγράμματος
76    printf ("=====\n\n");
77    printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
78    printf ("=====\n\n");
79    printf ("Εισάγετε N ακέραιους αριθμούς : ");
80    scanf ("%d", &N); // Εισαγωγή του πλήθους των ακεραίων αριθμών
81    printf ("\n-----\n\n");
82    printf ("Το πλήθος των ακεραίων : [%20d]\n\n", N);
83    // Εκτύπωση του πλήθους των ακεραίων αριθμών
84    if (N > 0) /* (~) Ένας ή περισσότεροι ακέραιοι αριθμοί */
85    {
86        while (i < N) /* Βρόχος */
87        {
88            printf ("-----\n\n");
89            printf ("Εισάγετε ακεραίο αριθμό : ");
90            scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών
91            printf ("\n-----\n\n");
92            printf ("Ο ακεραίος αριθμός : [%20d]\n\n", n);
93            // Εκτύπωση των ακεραίων αριθμών
94            if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
95            {
96                pow = n * n; // Υπολογισμός του τετραγώνου του
97                περιττού
98                printf ("Τετράγωνο περιττού : [%20d]\n\n",
99                pow); // Εκτύπωση του τετραγώνου του περιττού
100               printf ("Περιττός αριθμός\n");
101            }
102            else /* (!) Εισαγωγή άρτιου αριθμού */
103            {
104                evens++; // Υπολογισμός του πλήθους των άρτιων
105                αριθμών
106                printf ("\nΆρτιος αριθμός\n");
107            }
108            if (n >= 0) /* (+) Εισαγωγή θετικού αριθμού */
109            {
110                pos++; // Υπολογισμός του πλήθους των θετικών
111                αριθμών
112                sum = sum + n; // Υπολογισμός του αθροίσματος των
113                θετικών αριθμών
114                avg = (float) sum / pos; // Υπολογισμός του μέσου
115                όρου των θετικών αριθμών
116            }
117            else /* (+) Εισαγωγή αρνητικού αριθμού */
118            {
119                neg--; // Υπολογισμός του πλήθους των αρνητικών
120                αριθμών
121                sum = sum - n; // Υπολογισμός του αθροίσματος των
122                αρνητικών αριθμών
123                avg = (float) sum / neg; // Υπολογισμός του μέσου
124                όρου των αρνητικών αριθμών
125            }
126            i++;
127        }
128    }
129    printf ("\nΜέση τιμή : [%20f]\n\n", avg);
130    // Εκτύπωση του μέσου όρου
131    printf ("=====\n\n");
132    // Τίτλος του προγράμματος
133    printf ("=====\n\n");
134    printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
135    printf ("=====\n\n");
136    printf ("Εισάγετε N ακέραιους αριθμούς : ");
137    scanf ("%d", &N); // Εισαγωγή του πλήθους των ακεραίων αριθμών
138    printf ("\n-----\n\n");
139    printf ("Το πλήθος των ακεραίων : [%20d]\n\n", N);
140    // Εκτύπωση του πλήθους των ακεραίων αριθμών
141    if (N > 0) /* (~) Ένας ή περισσότεροι ακέραιοι αριθμοί */
142    {
143        while (i < N) /* Βρόχος */
144        {
145            printf ("-----\n\n");
146            printf ("Εισάγετε ακεραίο αριθμό : ");
147            scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών
148            printf ("\n-----\n\n");
149            printf ("Ο ακεραίος αριθμός : [%20d]\n\n", n);
150            // Εκτύπωση των ακεραίων αριθμών
151            if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
152            {
153                pow = n * n; // Υπολογισμός του τετραγώνου του
154                περιττού
155                printf ("Τετράγωνο περιττού : [%20d]\n\n",
156                pow); // Εκτύπωση του τετραγώνου του περιττού
157                printf ("Περιττός αριθμός\n");
158            }
159            else /* (!) Εισαγωγή άρτιου αριθμού */
160            {
161                evens++; // Υπολογισμός του πλήθους των άρτιων
162                αριθμών
163                printf ("\nΆρτιος αριθμός\n");
164            }
165            if (n >= 0) /* (+) Εισαγωγή θετικού αριθμού */
166            {
167                pos++; // Υπολογισμός του πλήθους των θετικών
168                αριθμών
169                sum = sum + n; // Υπολογισμός του αθροίσματος των
170                θετικών αριθμών
171                avg = (float) sum / pos; // Υπολογισμός του μέσου
172                όρου των θετικών αριθμών
173            }
174            else /* (+) Εισαγωγή αρνητικού αριθμού */
175            {
176                neg--; // Υπολογισμός του πλήθους των αρνητικών
177                αριθμών
178                sum = sum - n; // Υπολογισμός του αθροίσματος των
179                αρνητικών αριθμών
180                avg = (float) sum / neg; // Υπολογισμός του μέσου
181                όρου των αρνητικών αριθμών
182            }
183            i++;
184        }
185    }
186    printf ("\nΜέση τιμή : [%20f]\n\n", avg);
187    // Εκτύπωση του μέσου όρου
188    printf ("=====\n\n");
189    // Τίτλος του προγράμματος
190    printf ("=====\n\n");
191    printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
192    printf ("=====\n\n");
193    printf ("Εισάγετε N ακέραιους αριθμούς : ");
194    scanf ("%d", &N); // Εισαγωγή του πλήθους των ακεραίων αριθμών
195    printf ("\n-----\n\n");
196    printf ("Το πλήθος των ακεραίων : [%20d]\n\n", N);
197    // Εκτύπωση του πλήθους των ακεραίων αριθμών
198    if (N > 0) /* (~) Ένας ή περισσότεροι ακέραιοι αριθμοί */
199    {
200        while (i < N) /* Βρόχος */
201        {
202            printf ("-----\n\n");
203            printf ("Εισάγετε ακεραίο αριθμό : ");
204            scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών
205            printf ("\n-----\n\n");
206            printf ("Ο ακεραίος αριθμός : [%20d]\n\n", n);
207            // Εκτύπωση των ακεραίων αριθμών
208            if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
209            {
210                pow = n * n; // Υπολογισμός του τετραγώνου του
211                περιττού
212                printf ("Τετράγωνο περιττού : [%20d]\n\n",
213                pow); // Εκτύπωση του τετραγώνου του περιττού
214                printf ("Περιττός αριθμός\n");
215            }
216            else /* (!) Εισαγωγή άρτιου αριθμού */
217            {
218                evens++; // Υπολογισμός του πλήθους των άρτιων
219                αριθμών
220                printf ("\nΆρτιος αριθμός\n");
221            }
222            if (n >= 0) /* (+) Εισαγωγή θετικού αριθμού */
223            {
224                pos++; // Υπολογισμός του πλήθους των θετικών
225                αριθμών
226                sum = sum + n; // Υπολογισμός του αθροίσματος των
227                θετικών αριθμών
228                avg = (float) sum / pos; // Υπολογισμός του μέσου
229                όρου των θετικών αριθμών
230            }
231            else /* (+) Εισαγωγή αρνητικού αριθμού */
232            {
233                neg--; // Υπολογισμός του πλήθους των αρνητικών
234                αριθμών
235                sum = sum - n; // Υπολογισμός του αθροίσματος των
236                αρνητικών αριθμών
237                avg = (float) sum / neg; // Υπολογισμός του μέσου
238                όρου των αρνητικών αριθμών
239            }
240            i++;
241        }
242    }
243    printf ("\nΜέση τιμή : [%20f]\n\n", avg);
244    // Εκτύπωση του μέσου όρου
245    printf ("=====\n\n");
246    // Τίτλος του προγράμματος
247    printf ("=====\n\n");
248    printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
249    printf ("=====\n\n");
250    printf ("Εισάγετε N ακέραιους αριθμούς : ");
251    scanf ("%d", &N); // Εισαγωγή του πλήθους των ακεραίων αριθμών
252    printf ("\n-----\n\n");
253    printf ("Το πλήθος των ακεραίων : [%20d]\n\n", N);
254    // Εκτύπωση του πλήθους των ακεραίων αριθμών
255    if (N > 0) /* (~) Ένας ή περισσότεροι ακέραιοι αριθμοί */
256    {
257        while (i < N) /* Βρόχος */
258        {
259            printf ("-----\n\n");
260            printf ("Εισάγετε ακεραίο αριθμό : ");
261            scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών
262            printf ("\n-----\n\n");
263            printf ("Ο ακεραίος αριθμός : [%20d]\n\n", n);
264            // Εκτύπωση των ακεραίων αριθμών
265            if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
266            {
267                pow = n * n; // Υπολογισμός του τετραγώνου του
268                περιττού
269                printf ("Τετράγωνο περιττού : [%20d]\n\n",
270                pow
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
53          neg++; // Υπολογισμός του πλήθους των αρνητικών
αριθμών

54          prod = prod * n; // Υπολογισμός του γινομένου των
αρνητικών αριθμών
55      }
56      i++;
57  }

59      if (neg != 0) /* (-) Ένας ή περισσότεροι αρνητικοί αριθμοί */
60      {

61          printf ("-----
--\n\n");
62          printf ("Πλήθος άρτιων                : [%20d]\n", evens);
// Εκτύπωση του πλήθους των άρτιων αριθμών
63          printf ("Μέσος όρος θετικών                : [%20.6f]\n", avg);
// Εκτύπωση του μέσου όρου των θετικών αριθμών
64          printf ("Γινόμενο αρνητικών                : [%20d]\n", prod);
// Εκτύπωση του γινομένου των αρνητικών αριθμών
65      }
66      else /* (-) Κανένας αρνητικός αριθμός */
67      {
68          printf ("-----
--\n\n");
69          prod--; // Εκχώρηση τιμής "0" στη μεταβλητή "prod"
70          printf ("Πλήθος άρτιων                : [%20d]\n", evens);
// Εκτύπωση του πλήθους των άρτιων αριθμών
71          printf ("Μέσος όρος θετικών                : [%20.6f]\n", avg);
// Εκτύπωση του μέσου όρου των θετικών αριθμών
72          printf ("Γινόμενο αρνητικών                : [%20d]\n", prod);
// Εκτύπωση του γινομένου των αρνητικών αριθμών
73      }
74  }
75  else /* (~) Κανένας ακέραιος αριθμός */
76  {
77      printf      ("-----
\n\n");
78      prod--; // Εκχώρηση τιμής "0" στη μεταβλητή "prod"
79      printf ("Πλήθος άρτιων                : [%20d]\n", evens);
// Εκτύπωση του πλήθους των άρτιων αριθμών
80      printf ("Μέσος όρος θετικών                : [%20.6f]\n", avg);
// Εκτύπωση του μέσου όρου των θετικών αριθμών
81      printf ("Γινόμενο αρνητικών                : [%20d]\n", prod);
// Εκτύπωση του γινομένου των αρνητικών αριθμών
82  }
83
84  return 0;
85  }
```

ΤΕΚΜΗΡΙΩΣΗ «MathsWithIntegersInLoopA.c»

ΖΗΤΟΥΜΕΝΟ

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

a) Διαβάζει από τη «standard» είσοδο «N» ακέραιους αριθμούς.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

- b) Ελέγχει το πλήθος των ακεραίων που έδωσε ο χρήστης.
- c) Υπολογίζει το τετράγωνο των περιττών αριθμών.
- d) Υπολογίζει το πλήθος των άρτιων αριθμών.
- e) Υπολογίζει τον μέσο όρο των θετικών αριθμών.
- f) Υπολογίζει το γινόμενο των αρνητικών αριθμών.
- g) Τυπώνει τα αποτελέσματα συνοδευόμενα με τα κατάλληλα μηνύματα από την «standard» έξοδο.

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 ανάγνωση 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) αριθμητικοί : +, *, %
- b) σχεσιακοί : !=, >, <, ==
- c) ανάθεσης : =
- d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης «scanf()» που συνδέεται με τη «standard» είσοδο
- e) μετααύξησης : μεταβλητή++
- f) μεταμείωσης : μεταβλητή--

Οι εντολές ελέγχου :

- a) if – else

Οι εντολές επαναλήψης :

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

a) while

ΜΕΤΑΒΛΗΤΕΣ

Ακέραιες μεταβλητές (τύπου «int»)

N (Το πλήθος των ακεραίων αριθμών)

n (Ο κάθε ακέραιος αριθμός που εισάγεται μέσα στον βρόχο)

i (Μετρητής για τον έλεγχο του βρόχου)

pow (Το τετράγωνο των περιττών αριθμών)

evens (Το πλήθος των άρτιων αριθμών)

pos (Το πλήθος των θετικών αριθμών)

sum (Το άθροισμα των θετικών αριθμών)

neg (Το πλήθος των αρνητικών αριθμών)

prod (Το γινόμενο των αρνητικών αριθμών)

Πραγματικές μεταβλητές (τύπου «float»)

avg (Ο μέσος όρος των θετικών αριθμών)

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών (γραμμές 8-16)

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στις γραμμές «8-16» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Αρχικοποίηση μεταβλητών (γραμμές 9, 11, 12, 13, 14, 15)

Στις προαναφερόμενες γραμμές γίνονται μερικές αρχικοποιήσεις μεταβλητών.

Πιο αναλυτικά, στη γραμμή «9» καταχωρείται η τιμή «0» στην «i» για τον μετρητή που ελέγχει τον βρόχο, στη γραμμή «11» η τιμή «0» στην «evens» για το πλήθος των άρτιων αριθμών, στη γραμμή «12» η τιμή «0» στη «neg» για το πλήθος των αρνητικών αριθμών, στη γραμμή «13» η τιμή «1» στην «prod» για το γινόμενο των αρνητικών αριθμών, στη γραμμή «14» η τιμή

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

«0» στην «pos» για το πλήθος των θετικών αριθμών και στη γραμμή «15» η τιμή «0» στη «sum» για το άθροισμα των θετικών αριθμών.

Τίτλος προγράμματος **(γραμμή 19)**

Στην γραμμή «19» με μία συναρτήρηση «printf()» τυπώνεται από την «standard» έξοδο ο τίτλος του προγράμματος (Μαθηματικές πράξεις με ακέραιους αριθμούς), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

Εισαγωγή του πλήθους των ακεραίων αριθμών **(γραμμή 22)**

Στη γραμμή «22» με τη συνάρτηση «scanf()» διαβάζεται από τη «standard» είσοδο το πλήθος των ακεραίων αριθμών που θα εισαχθούν. Με τη συνάρτηση «printf()» τυπώνεται από τη «standard» έξοδο το αντίστοιχο μήνυμα (γραμμή 21).

Εκτύπωση του πλήθους των ακεραίων αριθμών **(γραμμή 24)**

Στη γραμμή «24» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «N» (το πλήθος των ακεραίων αριθμών που θα εισαχθούν) 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.

(~) Ένας ή περισσότεροι ακέραιοι αριθμοί **(γραμμές 25-75)** **(~) Κανένας ακέραιος αριθμός** **(γραμμές 75-82)**

Στις γραμμές «25-82» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για το πλήθος των ακεραίων αριθμών που εισάγει ο χρήστης.

Διεξοδικά, αν ο χρήστης εισάγει έναν ή περισσότερους ακέραιους αριθμούς, τότε, θα εκτελεστούν οι εντολές της υποενότητας «Ένας ή περισσότεροι ακέραιοι αριθμοί» στις γραμμές «25-75», διαφορετικά αν δεν εισάγει κανέναν ακέραιο αριθμό, τότε, θα εκτελεστούν οι εντολές της υποενότητας «Κανένας ακέραιος αριθμός» στις γραμμές «75-82».

(~) Ένας ή περισσότεροι ακέραιοι αριθμοί **(γραμμές 25-75)**

Στις γραμμές «25-75» εκτελούνται οι εντολές της «if» της γραμμή «25» για την περίπτωση που η παράσταση «N > 0» παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει τουλάχιστον έναν ακέραιο αριθμό).

Βρόχος **(γραμμές 27-57)**

Στις γραμμές «27-57» εκτελείται με μία εντολή επανάληψης «while» ένας

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

βρόχος για την εισαγωγή των ακεραίων αριθμών από τη «standard» είσοδο και για τον υπολογισμό των επιθυμητών αποτελεσμάτων (τετράγωνο περιττού, πλήθος άρτιων, μέσος όρος θετικών και γινόμενο αρνητικών αριθμών).

Ο βρόχος εκτελείται για όσο η τιμή της παράστασης « $i < N$ » παράγει αποτέλεσμα μία τιμή «True». Με άλλα λόγια, ο βρόχος εκτελείται μέχρι ο χρήστης να εισάγει όλους τους ακέραιους αριθμούς που επέλεξε να εισάγει. Η μεταβλητή « i » που χρησιμοποιείται ως βοηθητική συνάρτηση για τον έλεγχο του βρόχου έχει ως αρχική τιμή την τιμή «0» που καταχωρήθηκε στη δήλωση των μεταβλητών (αναλυτικά βλ. υποενότητες «Δήλωση μεταβλητών (γραμμές 8-16)», «Αρχικοποίηση μεταβλητών (γραμμές 9)») και στην τελευταία εντολή του βρόχου (γραμμή «56») το περιεχόμενο της αυξάνεται κατά «1» με την παράσταση « $i++$ ».

Εισαγωγή των ακεραίων αριθμών	(γραμμή 31)
-------------------------------	-------------

Στη γραμμή «31» διαβάζεται από τη «standard» είσοδο με τη συνάρτηση «scanf()», ο ακέραιος αριθμός που εισάγει ο χρήστης συνοδευόμενο με το κατάλληλο μήνυμα που τυπώνεται από την «standard» έξοδο με την συνάρτηση «printf()» (γραμμή 30) κάθε φορά που εκτελείται ο βρόχος.

Εκτύπωση των ακεραίων αριθμών	(γραμμή 32)
-------------------------------	-------------

Στη γραμμή «32» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής « n » (ο ακέραιος αριθμός που εισάγει ο χρήστης κάθε φορά που εκτελείται ο βρόχος) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

(!) Εισαγωγή περιττού αριθμού	(γραμμές 34-39)
-------------------------------	-----------------

(!) Εισαγωγή άρτιου αριθμού	(γραμμές 40-44)
-----------------------------	-----------------

Στις γραμμές «34-44» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής περιττού ή άρτιου αριθμού.

Διεξοδικά, αν ο χρήστης εισάγει περιττό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή περιττού αριθμού» στις γραμμές «34-39», διαφορετικά αν εισάγει άρτιο αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή άρτιου αριθμού» στις γραμμές «40-44».

(!) Εισαγωγή περιττού αριθμού	(γραμμές 34-39)
-------------------------------	-----------------

Στις γραμμές «34-39» εκτελούνται οι εντολές της «if» της γραμμή «34» για την περίπτωση που η παράσταση « $n \% 2 != 0$ » παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει περιττό αριθμό).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Υπολογισμός του τετραγώνου του περιττού (γραμμή 36)

Στη γραμμή «36» υπολογίζεται με την αριθμητική παράσταση « $n * n$ » η ύψωση της δύναμης του δοθέντος περιττού αριθμού εις το τετράγωνο και το αποτέλεσμα καταχωρείται στη μεταβλητή «row».

Εκτύπωση του τετραγώνου του περιττού (γραμμή 37)

Στη γραμμή «37» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «row» (η ύψωση της δύναμης του δοθέντος περιττού αριθμού εις το τετράγωνο) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

(!) Εισαγωγή άρτιου αριθμού (γραμμές 40-44)

Στις γραμμές «40-44» εκτελούνται οι εντολές της «else» (γραμμή 40) που αντιστοιχούν στην «if» της γραμμής «34» για την περίπτωση που η παράσταση στην « $n \% 2 != 0$ » παράγει αποτέλεσμα μία τιμή «False» (ο χρήστης εισάγει άρτιο αριθμό).

Υπολογισμός του πλήθους των άρτιων αριθμών (γραμμή 42)

Στη γραμμή «42» υπολογίζεται με την αριθμητική παράσταση «evens++» (evens = evens + 1) το πλήθος των άρτιων αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή «evens» που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 11)»).

(+) Εισαγωγή θετικού αριθμού (γραμμές 45-50)

(+) Εισαγωγή αρνητικού αριθμού (γραμμές 51-55)

Στις γραμμές «45-55» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής θετικού αριθμού ή αρνητικού αριθμού.

Διεξοδικά, αν ο χρήστης εισάγει θετικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή θετικού αριθμού» στις γραμμές «45-50», διαφορετικά αν εισάγει αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή αρνητικού αριθμού» στις γραμμές «51-55».

(+) Εισαγωγή θετικού αριθμού (γραμμές 45-50)

Στις γραμμές «45-50» εκτελούνται οι εντολές της «if» της γραμμής «45»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

για την περίπτωση που η παράσταση « $n \geq 0$ » παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει θετικό αριθμό).

Υπολογισμός του πλήθους των θετικών αριθμών (γραμμή 47)

Στη γραμμή «47» υπολογίζεται με την αριθμητική παράσταση « $ros++$ » ($ros = ros + 1$) το πλήθος των θετικών αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή « ros » που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 14)»).

Υπολογισμός του αθροίσματος των θετικών αριθμών (γραμμή 48)

Στη γραμμή «48» υπολογίζεται με την αριθμητική παράσταση « $sum + n$ » το άθροισμα των δοθέντων θετικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή « sum » που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 15)»).

Υπολογισμός του μέσου όρου των θετικών αριθμών (γραμμή 49)

Στη γραμμή «49» υπολογίζεται με την αριθμητική παράσταση « $(float) sum / ros$ » ο μέσος όρος των θετικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή « avg ». Επειδή, από τη « $standard$ » είσοδο διαβάζονται ακέραιοι αριθμοί και η μεταβλητή « avg » που υπολογίζει τον μέσο όρο των θετικών, είναι πραγματική ($float$), μετατρέπονται οι ακέραιες μεταβλητές « sum » και « ros » σε πραγματικές μόνο για την αριθμητική παράσταση της γραμμής «49».

(+) Εισαγωγή αρνητικού αριθμού (γραμμές 51-55)

Στις γραμμές «51-55» εκτελούνται οι εντολές της « $else$ » (γραμμή 51) που αντιστοιχεί στην « if » της γραμμής «45», για την περίπτωση που η παράσταση « $n \geq 0$ » παράγει αποτέλεσμα την τιμή «False» (ο χρήστης εισάγει αρνητικό αριθμό).

Υπολογισμός του πλήθους των αρνητικών αριθμών (γραμμή 53)

Στη γραμμή «53» υπολογίζεται με την αριθμητική παράσταση « $neg++$ » ($neg = neg + 1$) το πλήθος των αρνητικών αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή « neg » που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 12)»).

Υπολογισμός του γινομένου των αρνητικών αριθμών (γραμμή 54)

Στη γραμμή «54» υπολογίζεται με την αριθμητική παράσταση « $prod * n$ » το άθροισμα των δοθέντων αρνητικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή « $prod$ » που έχει για αρχική τιμή, την τιμή «1» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 13)»).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

(-) Ένας ή περισσότεροι αρνητικοί αριθμοί (γραμμές 59-65)
(-) Κανένας αρνητικός αριθμός (γραμμές 66-73)

Στις γραμμές «59-73» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής αρνητικών αριθμών για την ορθότερη αποτύπωση των επιθυμητών αποτελεσμάτων.

Διεξοδικά, αν ο χρήστης εισάγει έστω έναν αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Ένας ή περισσότεροι αρνητικοί αριθμοί» στις γραμμές «59-65», διαφορετικά αν δεν εισάγει αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Κανένας αρνητικός αριθμός» στις γραμμές «66-73».

(-) Ένας ή περισσότεροι αρνητικοί αριθμοί (γραμμές 59-65)

Στις γραμμές «59-65» εκτελούνται οι εντολές της «if» της γραμμής «59» για την περίπτωση που η παράσταση «neg != 0» παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει τουλάχιστον έναν αρνητικό αριθμό).

Εκτύπωση του πλήθους των άρτιων αριθμών (γραμμή 62)

Στη γραμμή «62» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «evens» (το πλήθος των άρτιων αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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 μέσου όρου των θετικών αριθμών (γραμμή 63)

Στη γραμμή «63» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «avg» (ο μέσος όρος των θετικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6f», 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 γινομένου των αρνητικών αριθμών (γραμμή 64)

Στη γραμμή «64» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «prod» (το γινόμενο των αρνητικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιόμο-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ρφη στοίχιση των αποτελεσμάτων.

(-) Κανένας αρνητικός αριθμός (γραμμές 66-73)

Στις γραμμές «66-73» εκτελούνται οι εντολές της «else» (γραμμή 66) που αντιστοιχεί στην «if» της γραμμής «59», για την περίπτωση που η παράσταση «n >= 0» παράγει αποτέλεσμα την τιμή «False» (ο χρήστης δεν εισάγει αρνητικό αριθμό).

Εκχώρηση τιμής "0" στη μεταβλητή "prod" (γραμμή 69)

Στη γραμμή «69» μειώνεται το περιεχόμενο της μεταβλητής «prod» κατά «1» με την παράσταση «prod- -», όπου έχει αρχικά την τιμή «1» από την δήλωση μεταβλητών (αναλυτικά βλ. υποενότητα «Αρχικοποίηση μεταβλητών (γραμμή 13)»), ώστε να αποτυπωθεί το αποτέλεσμα «Γινόμενο αρνητικών : 0» από την στιγμή που δεν υπάρχουν αρνητικοί και όχι ψευδώς το αποτέλεσμα «Γινόμενο αρνητικών : 1».

Εκτύπωση του πλήθους των άρτιων αριθμών (γραμμή 70)

Στη γραμμή «70» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «evens» (το πλήθος των άρτιων αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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 μέσου όρου των θετικών αριθμών (γραμμή 71)

Στη γραμμή «71» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «avg» (ο μέσος όρος των θετικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6f», 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 γινομένου των αρνητικών αριθμών (γραμμή 72)

Στη γραμμή «72» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «prod» (το γινόμενο των αρνητικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιό-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

μορφή στοίχιση των αποτελεσμάτων.

(~) Κανέναν ακέραιο αριθμός

(γραμμές 75-82)

Στις γραμμές «75-82» εκτελούνται οι εντολές της «else» (γραμμή 75) που αντιστοιχεί στην «if» της γραμμής «25», για την περίπτωση που η παράσταση « $N > 0$ » παράγει αποτέλεσμα την τιμή «False» (ο χρήστης δεν εισάγει κανέναν ακέραιο αριθμό ή λανθασμένα δίνει αρνητική τιμή στο πλήθος των ακεραίων αριθμών).

Εκχώρηση τιμής "0" στη μεταβλητή "prod"

(γραμμή 78)

Στη γραμμή «78» μειώνεται το περιεχόμενο της μεταβλητής «prod» κατά «1» με την παράσταση «prod- -», όπου έχει αρχικά την τιμή «1» από την δήλωση μεταβλητών (αναλυτικά βλ. υποενότητα «Αρχικοποίηση μεταβλητών (γραμμή 13)»), ώστε να αποτυπωθεί το αποτέλεσμα «Γινόμενο αρνητικών : 0» από την στιγμή που δεν υπάρχουν αρνητικοί και όχι ψευδώς το αποτέλεσμα «Γινόμενο αρνητικών : 1».

Εκτύπωση του πλήθους των άρτιων αριθμών

(γραμμή 79)

Στη γραμμή «70» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «evens» (το πλήθος των άρτιων αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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 μέσου όρου των θετικών αριθμών

(γραμμή 80)

Στη γραμμή «71» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «avg» (ο μέσος όρος των θετικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6f», 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 γινομένου των αρνητικών αριθμών

(γραμμή 81)

Στη γραμμή «72» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «prod» (το γινόμενο των αρνητικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20d», δηλαδή, το αποτέλεσμα θα τυπωθεί αμέ-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

σως μετά από «20» κενούς χαρακτήρες. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1 ($N > 0$)

=====

Μαθηματικές πράξεις με ακέραιους αριθμούς

=====

Εισάγετε N ακέραιους αριθμούς : 6

Το πλήθος των ακεραιων : [6]

Εισάγετε ακέραιο αριθμό : 8

Ο ακέραιος αριθμός : [8]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : -11

Ο ακέραιος αριθμός : [-11]

Τετράγωνο περιττού : [121]

Περιττός αριθμός

Εισάγετε ακέραιο αριθμό : 0

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο ακέραιος αριθμός : [0]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : 57

Ο ακέραιος αριθμός : [57]

Τετράγωνο περιττού : [3249]

Περιττός αριθμός

Εισάγετε ακέραιο αριθμό : -12

Ο ακέραιος αριθμός : [-12]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : 5

Ο ακέραιος αριθμός : [5]

Τετράγωνο περιττού : [25]

Περιττός αριθμός

Πλήθος άρτιων : [3]

Μέσος όρος θετικών : [17.500000]

Γινόμενο αρνητικών : [132]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Παράδειγμα 2 ($N < 0$ ή $N == 0$)

=====

Μαθηματικές πράξεις με ακέραιους αριθμούς

=====

Εισάγετε N ακέραιους αριθμούς : 0

Το πλήθος των ακέραιων : [0]

Πλήθος άρτιων : [0]

Μέσος όρος θετικών : [0.000000]

Γινόμενο αρνητικών : [0]

ΠΑΡΑΤΗΡΗΣΕΙΣ

Η ορθότητα των αποτελεσμάτων στα παραπάνω παραδείγματα ελέγχθηκε (με μία μικρή παρατήρηση) και με την χρήση τυπικής αριθμομηχανής και δεν διαπίστωθηκε καμία διαφορά.

Αναλυτικά, έχουμε δύο παραδείγματα όπου το καθένα αντιστοιχεί σε ειδικές συνθήκες που βρίσκονται μέσα στην αντίστοιχη εντολή «if ($N > 0$)» (γραμμή 22) και της «else» (γραμμή 71) :

Παράδειγμα 1 ($N > 0$)

Αντιπροσωπεύει την περίπτωση όπου ο χρήστης εισάγει έναν ή περισσότερους ακέραιους αριθμούς.

Αναλυτικά, ο χρήστης εισάγει «6» ακέραιους αριθμούς εκ των οποίων οι «3» είναι άρτιοι (8, 0, -12) και οι άλλοι «3» περιττοί (-11, 57, 5). Από αυτούς οι «4» είναι θετικοί (0, 5, 8, 57) και οι «2» αρνητικοί (-11, -12).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Η σχετική παρατήρηση είναι ότι στο συγκεκριμένο πρόγραμμα ο αριθμός «0» θεωρείται ως άρτιος και θετικός. Τα αποτελέσματα :

Τετράγωνο περιττού «-11» :	121	$\{(-11)^2\}$
Τετράγωνο περιττού «57» :	3249	(57^2)
Τετράγωνο περιττού «5» :	25	(5^2)
Πλήθος άρτων :	3	$(8, 0, -12)$
Μέσος όρος θετικών :	17.500000	$\{(0 + 5 + 8 + 57) / 4\}$
Γινόμενο αρνητικών :	132	$(-11 * -12)$

Παράδειγμα 2 (N < 0 ή N == 0)

Απεικονίζει την περίπτωση όπου ο χρήστης δεν εισάγει κανέναν ακέραιο αριθμό ή πληκτρολογεί καταλάθος αρνητική τιμή στο πλήθος των ακεραίων αριθμών που θέλει να εισάγει.

Σαφώς, δεν υπάρχουν άρτιοι, περιττοί, θετικοί, αρνητικοί αριθμοί. Τα αποτελέσματα :

Πλήθος αρτίων :	0
Μέσος όρος θετικών :	0.000000
Γινόμενο αρνητικών :	0

ΘΕΜΑ 3B

ΕΠΙΣΗΜΑΝΣΗ «MathsWithIntegersInLoopB.c»

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

Πιο αναλυτικά, στην περίπτωση όπου ο χρήστης εισάγει όσους ακεραίους θέλει και ο βρόχος σταματάει μέχρι ο χρήστης να εισάγει την τιμή «0».

ΠΡΟΓΡΑΜΜΑ «MathsWithIntegersInLoopB.c»

```
1 #include <stdio.h>
2
3 int main (int argc, int **argv)
4
5 {
6     system ("chcp 1253");
7
```


ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
8  int n; // Δήλωση μεταβλητών

9  int pow;

10 int evens = 0; // Αρχικοποίηση μεταβλητών

11 int neg = 0; // Αρχικοποίηση μεταβλητών

12 int prod = 1; // Αρχικοποίηση μεταβλητών

13 int pos = 0; // Αρχικοποίηση μεταβλητών

14 int sum = 0; // Αρχικοποίηση μεταβλητών

15 float avg;

16

17 printf ("=====\n\n");

18 printf ("Μαθηματικές πράξεις με ακέραιους αριθμούς\n\n");
// Τίτλος του προγράμματος

19 printf ("=====\n\n");

20 do /* Βρόχος */
21 {
22     printf ("-----\n\n");

23     printf ("Εισάγετε ακέραιο αριθμό : ");

24     scanf ("%d", &n); // Εισαγωγή των ακεραίων αριθμών

25     printf ("\n-----\n\n");

26     printf ("Ο ακέραιος αριθμός          : [%20d]\n\n", n);
// Εκτύπωση των ακεραίων αριθμών

27     if (n != 0) /* (~) Εισαγωγή τιμής διάφορης του "0" */
28     {
29         if (n % 2 != 0) /* (!) Εισαγωγή περιττού αριθμού */
30         {
31             pow = n * n; // Υπολογισμός του τετραγώνου του
περιττού

32             printf ("Τετράγωνο περιττού      : [%20d]\n\n", pow);
// Εκτύπωση του τετραγώνου του περιττού

33             printf ("\nΠεριττός αριθμός\n");

34         }
35         else /* (!) Εισαγωγή άρτιου αριθμού */
36         {
37             evens++; // Υπολογισμός του πλήθους των άρτιων
αριθμών

38             printf ("\nΆρτιος αριθμός\n");
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
39         }
40         if (n > 0) /* (+) Εισαγωγή θετικού αριθμού */
41         {
42             pos++; // Υπολογισμός του πλήθους των θετικών
αριθμών
43             sum = sum + n; // Υπολογισμός του αθροίσματος των
θετικών αριθμών
44             avg = (float) sum / pos; // Υπολογισμός του μέσου
όρου των θετικών αριθμών
45         }
46         else /* (+) Εισαγωγή αρνητικού αριθμού */
47         {
48             neg++; // Υπολογισμός του πλήθους των αρνητικών
αριθμών
49             prod = prod * n; // Υπολογισμός του γινομένου των
αρνητικών αριθμών
50         }
51     }
52     else /* (~) Εισαγωγή της τιμής "0" */
53     {
54         printf ("\n~Τέλος βρόχου~\n");
55     }
56 }
57 while (n != 0);
58
59 if (neg != 0) /* (-) Ένας ή περισσότεροι αρνητικοί αριθμοί */
60 {
61     printf ("-----
\n\n");
62     printf ("Πλήθος άρτιων                : [%20d]\n",  evens);
// Εκτύπωση του πλήθους των άρτιων αριθμών
63     printf ("Μέσος όρος θετικών                : [%20.6f]\n",  avg);
// Εκτύπωση του μέσου όρου των θετικών αριθμών
64     printf ("Γινόμενο αρνητικών                : [%20d]\n",  prod);
// Εκτύπωση του γινομένου των αρνητικών αριθμών
65 }
66 else /* (-) Κανένας αρνητικός αριθμός */
67 {
68     printf ("-----
\n\n");
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
69      prod--; // Εκχώρηση τιμής "0" στη μεταβλητή "prod"
70      printf ("Πλήθος άρτιων                : [%20d]\n",  evens);
// Εκτύπωση του πλήθους των άρτιων αριθμών
71      printf ("Μέσος όρος θετικών            : [%20.6f]\n",  avg);
// Εκτύπωση του μέσου όρου των θετικών αριθμών
72      printf ("Γινόμενο αρνητικών           : [%20d]\n",  prod);
// Εκτύπωση του γινομένου των αρνητικών αριθμών
73  }
74
75  return 0;
76  }
```

ΤΕΚΜΗΡΙΩΣΗ «MathsWithIntegersInLoopB.c»

ZΗΤΟΥΜΕΝΟ

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

- a) Διαβάζει από τη «standard» είσοδο «N» ακέραιους αριθμούς μέχρι να διαβάσει τον αριθμό «0».
- b) Υπολογίζει το τετράγωνο των περιττών αριθμών.
- c) Υπολογίζει το πλήθος των άρτιων αριθμών.
- d) Υπολογίζει τον μέσο όρο των θετικών αριθμών.
- e) Υπολογίζει το γινόμενο των αρνητικών αριθμών.
- f) Τυπώνει τα επιθυμητά αποτελέσματα από τη «standard» έξοδο συνοδευόμενα με τα κατάλληλα μηνύματα.

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 ανάγνωση και τηντύπωση περιεχομένων των αντίστοιχων μεταβλητών. Επίσης, η «printf()» χρησιμοποιήθηκε για να τυπωθούν χαρακτηριστικά μηνύματα για την βέλτιστη κατανόηση του πηγαίου κώδικα.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Επιπρόσθετα, χρησιμοποιήθηκαν οι χαρακτηριστικοί τελεστές :

a) αριθμητικοί : +, *, %

b) σχεσιακοί : !=, >, <, ==

c) ανάθεσης : =

d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης «scanf()» που συνδέεται με τη «standard» είσοδο

e) μετααύξησης : μεταβλητή++

f) μεταμείωσης : μεταβλητή--

Οι εντολές ελέγχου :

a) if – else

Οι εντολές επαναλήψης :

a) do - while

ΜΕΤΑΒΛΗΤΕΣ

Ακέραιες μεταβλητές (τύπου «int»)

n (Ο κάθε ακέραιος αριθμός που εισάγεται μέσα στον βρόχο)

row (Το τετράγωνο των περιττών αριθμών)

evens (Το πλήθος των άρτιων αριθμών)

pos (Το πλήθος των θετικών αριθμών)

sum (Το άθροισμα των θετικών αριθμών)

neg (Το πλήθος των αρνητικών αριθμών)

prod (Το γινόμενο των αρνητικών αριθμών)

Πραγματικές μεταβλητές (τύπου «float»)

avg (Ο μέσος όρος των θετικών αριθμών)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών

(γραμμές 8-15)

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

Αρχικοποίηση μεταβλητών

(γραμμές 10, 11, 12, 13, 14)

Στις προαναφερόμενες γραμμές γίνονται μερικές αρχικοποιήσεις μεταβλητών. Πιο αναλυτικά, στη γραμμή «10» η τιμή «0» στην «evens» για το πλήθος των άρτιων αριθμών, στην γραμμή «11» η τιμή «0» στη «neg» για το πλήθος των αρνητικών αριθμών, στη γραμμή «12» η τιμή «1» στην «prod» για το γινόμενο των αρνητικών αριθμών, στη γραμμή «13» η τιμή «0» στην «pos» για το πλήθος των θετικών αριθμών και στη γραμμή «14» η τιμή «0» στην «sum» για το άθροισμα των θετικών αριθμών.

Τίτλος προγράμματος

(γραμμή 18)

Στη γραμμή «18» με μία συναρτήση «printf()» τυπώνεται από την «standard» έξοδο ο τίτλος του προγράμματος (Μαθηματικές πράξεις με ακέραιους αριθμούς), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

Βρόχος

(γραμμές 20-57)

Στις γραμμές «20-57» εκτελείται με μία εντολή επανάληψης «do - while» ένας βρόχος για την εισαγωγή των ακεραίων αριθμών από τη «standard» είσοδο και για τον υπολογισμό των επιθυμητών αποτελεσμάτων (τετράγωνο περιττού, πλήθος άρτιων, μέσος όρος θετικών και γινόμενο αρνητικών αριθμών).

Ο βρόχος θα εκτελεστεί τουλάχιστον μία φορά, ανεξάρτητα τι τιμή παράγει η παράσταση που την ελέγχει και εκτελείται για όσο η τιμή της παράστασης «n != 0» παράγει αποτέλεσμα μία τιμή «True». Με άλλα λόγια, ο βρόχος εκτελείται μέχρι ο χρήστης εισάγει τον αριθμό «0». Ο έλεγχος της παράστασης που ελέγχει τον βρόχο πραγματοποιείται στη γραμμή «57»

Εισαγωγή των ακεραίων αριθμών

(γραμμή 24)

Στη γραμμή «24» διαβάζεται από τη «standard» είσοδο με τη συνάρτηση «scanf()», ο ακέραιος αριθμός που εισάγει ο χρήστης συνοδευόμενο με το κατάλληλο μήνυμα που τυπώνεται από την «standard» έξοδο με την συνάρτηση «printf()» (γραμμή 23) κάθε φορά που εκτελείται ο βρόχος.

Εκτύπωση των ακεραίων αριθμών

(γραμμή 26)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Στη γραμμή «26» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «n» (ο ακέραιος αριθμός που εισάγει ο χρήστης κάθε φορά που εκτελείται ο βρόχος) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

(~) Εισαγωγή τιμής διάφορης του «0»	(γραμμές 27-51)
(~) Εισαγωγή της τιμής «0»	(γραμμές 52-55)

Στις γραμμές «27-55» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής του αριθμού «0» που τερματίζει τον βρόχο.

Διεξοδικά, αν ο χρήστης δεν εισάγει τον αριθμό «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή τιμής διάφορης του “0”» στις γραμμές «27-51», διαφορετικά αν εισάγει τον αριθμό «0», τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή της τιμής “0”» στις γραμμές «52-55».

(~) Εισαγωγή τιμής διάφορης του «0»	(γραμμές 27-51)
-------------------------------------	-----------------

Στις γραμμές «27-51» εκτελούνται οι εντολές της «if» της γραμμής «27» ($n \neq 0$) (γραμμή 23) για την περίπτωση που η παράσταση « $n \neq 0$ » παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης δεν εισάγει τον αριθμό «0»).

(!) Εισαγωγή περιττού αριθμού	(γραμμές 29-34)
(!) Εισαγωγή άρτιου αριθμού	(γραμμές 35-39)

Στις γραμμές «29-39» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής περιττού ή άρτιου αριθμού.

Διεξοδικά, αν ο χρήστης εισάγει περιττό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή περιττού αριθμού» στις γραμμές «29-34», διαφορετικά αν εισάγει άρτιο αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή άρτιου αριθμού» στις γραμμές «35-39».

(!) Εισαγωγή περιττού αριθμού	(γραμμές 29-34)
-------------------------------	-----------------

Στις γραμμές «29-34» εκτελούνται οι εντολές της «if» της γραμμής «29» για την περίπτωση που η παράσταση « $n \% 2 \neq 0$ » παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει περιττό αριθμό).

Υπολογισμός του τετραγώνου του περιττού	(γραμμή 31)
---	-------------

Στη γραμμή «31» υπολογίζεται με την αριθμητική παράσταση « $n * n$ » η

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ύψωση της δύναμης του δοθέντος περιττού αριθμού εις το τετράγωνο και το αποτέλεσμα καταχωρείται στη μεταβλητή «row».

Εκτύπωση του τετραγώνου του περιττού (γραμμή 32)

Στη γραμμή «32» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «row» (η ύψωση της δύναμης του δοθέντος περιττού αριθμού εις το τετράγωνο) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

(!) Εισαγωγή άρτιου αριθμού (γραμμές 35-39)

Στις γραμμές «35-39» εκτελούνται οι εντολές της «else» (γραμμή 35) που αντιστοιχούν στην «if» της γραμμής «29» για την περίπτωση που η παράσταση στην «n % 2 != 0» παράγει αποτέλεσμα μία τιμή «False» (ο χρήστης εισάγει άρτιο αριθμό).

Υπολογισμός του πλήθους των άρτιων αριθμών (γραμμή 37)

Στη γραμμή «37» υπολογίζεται με την αριθμητική παράσταση «evens++» (evens = evens + 1) το πλήθος των άρτιων αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή «evens» που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 10)»).

(+) Εισαγωγή θετικού αριθμού (γραμμές 40-45)

(+) Εισαγωγή αρνητικού αριθμού (γραμμές 46-50)

Στις γραμμές «40-50» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής θετικού ή αρνητικού αριθμού.

Διεξοδικά, αν ο χρήστης εισάγει θετικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή θετικού αριθμού» στις γραμμές «40-45», διαφορετικά αν εισάγει αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Εισαγωγή αρνητικού αριθμού» στις γραμμές «46-50».

(+) Εισαγωγή θετικού αριθμού (γραμμές 40-45)

Στις γραμμές «40-45» εκτελούνται οι εντολές της «if» της γραμμής «40» για την περίπτωση που η παράσταση «n > 0» παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει θετικό αριθμό).

Υπολογισμός του πλήθους των θετικών αριθμών (γραμμή 42)

Στη γραμμή «42» υπολογίζεται με την αριθμητική παράσταση «pos++»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

($ros = ros + 1$) το πλήθος των θετικών αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή «ros» που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 13)»).

Υπολογισμός του αθροίσματος των θετικών αριθμών (γραμμή 43)

Στη γραμμή «43» υπολογίζεται με την αριθμητική παράσταση «sum + n» το άθροισμα των δοθέντων θετικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή «sum» που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 14)»).

Υπολογισμός του μέσου όρου των θετικών αριθμών (γραμμή 44)

Στη γραμμή «44» υπολογίζεται με την αριθμητική παράσταση «(float) sum / ros» ο μέσος όρος των θετικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή «avg». Επειδή, από τη «standard» είσοδο διαβάζονται ακέραιοι αριθμοί και η μεταβλητή «avg» που υπολογίζει τον μέσο όρο των θετικών, είναι πραγματική (float), μετατρέπονται οι ακέραιες μεταβλητές «sum» και «ros» σε πραγματικές μόνο για την αριθμητική παράσταση της γραμμής «44».

(+) Εισαγωγή αρνητικού αριθμού (γραμμές 46-50)

Στις γραμμές «46-50» εκτελούνται οι εντολές της «else» (γραμμή 46) που αντιστοιχεί στην «if» της γραμμής «40», για την περίπτωση που η παράσταση «n > 0» παράγει αποτέλεσμα την τιμή «False» (ο χρήστης εισάγει αρνητικό αριθμό).

Υπολογισμός του πλήθους των αρνητικών αριθμών (γραμμή 48)

Στη γραμμή «48» υπολογίζεται με την αριθμητική παράσταση «neg++» ($neg = neg + 1$) το πλήθος των αρνητικών αριθμών που έχει εισάγει ο χρήστης και το αποτέλεσμα καταχωρείται στη μεταβλητή «neg» που έχει για αρχική τιμή, την τιμή «0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 11)»).

Υπολογισμός του γινομένου των αρνητικών αριθμών (γραμμή 49)

Στη γραμμή «49» υπολογίζεται με την αριθμητική παράσταση «prod * n» το άθροισμα των δοθέντων αρνητικών αριθμών και το αποτέλεσμα καταχωρείται στη μεταβλητή «prod» που έχει για αρχική τιμή, την τιμή «1» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 12)»).

(~) Εισαγωγή της τιμής «0» (γραμμές 52-55)

Στις γραμμές «52-55» εκτελούνται οι εντολές της «else» (γραμμή 52) που

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

αντιστοιχεί στην «if» της γραμμής «27» για την περίπτωση που η παράσταση «n != 0» παράγει αποτέλεσμα την τιμή «False» (ο χρήστης εισάγει τον αριθμό «0»).

Πιο αναλυτικά, ελέγχει αν η ο χρήστης έχει καταχωρήσει την τιμή «0» στη μεταβλητή «n» και αν η παράσταση «n != 0» παράγει αποτέλεσμα την τιμή «False», τότε τυπώνεται από τη «standard» έξοδο με μία συνάρτηση «printf()» το μήνυμα «~Τέλος βρόχου~» με δύο χαρακτήρες διαφυγής της αλλαγής γραμμής (\n) στη γραμμή «54» επισημαίνοντας το τέλος του βρόχου, καθώς ο χρήστης πληκτρολόγησε την τερματική τιμή «0».

(-) Ένας ή περισσότεροι αρνητικοί αριθμοί (γραμμές 59-65) **(-) Κανένας αρνητικός αριθμός (γραμμές 66-73)**

Στις γραμμές «59-73» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση εισαγωγής αρνητικών αριθμών για την ορθότερη αποτύπωση των επιθυμητών αποτελεσμάτων.

Διεξοδικά, αν ο χρήστης εισάγει έστω έναν αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Ένας ή περισσότεροι αρνητικοί αριθμοί» στις γραμμές «59-65», διαφορετικά αν δεν εισάγει αρνητικό αριθμό, τότε θα εκτελεστούν οι εντολές της υποενότητας «Κανένας αρνητικός αριθμός» στις γραμμές «66-73».

(-) Ένας ή περισσότεροι αρνητικοί αριθμοί (γραμμές 59-65)

Στις γραμμές «59-65» εκτελούνται οι εντολές της «if» της γραμμής «59» για την περίπτωση που η παράσταση «neg != 0» παράγει αποτέλεσμα μία τιμή «True» (ο χρήστης εισάγει τουλάχιστον έναν αρνητικό αριθμό).

Εκτύπωση του πλήθους των άρτιων αριθμών (γραμμή 62)

Στη γραμμή «62» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «evens» (το πλήθος των άρτιων αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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 μέσου όρου των θετικών αριθμών (γραμμή 63)

Στη γραμμή «63» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «avg» (ο μέσος όρος των θετικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6f», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του γινομένου των αρνητικών αριθμών (γραμμή 64)

Στη γραμμή «64» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «prod» (το γινόμενο των αρνητικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

(-) Κανένας αρνητικός αριθμός (γραμμές 66-73)

Στις γραμμές «66-73» εκτελούνται οι εντολές της «else» (γραμμή 66) που αντιστοιχεί στην «if» της γραμμής «59», για την περίπτωση που η παράσταση «n >= 0» παράγει αποτέλεσμα την τιμή «False» (ο χρήστης δεν εισάγει αρνητικό αριθμό).

Εκχώρηση τιμής "0" στη μεταβλητή "prod" (γραμμή 69)

Στη γραμμή «69» μειώνεται το περιεχόμενο της μεταβλητής «prod» κατά «1» με την παράσταση «prod- -», όπου έχει αρχικά την τιμή «1» από την δήλωση μεταβλητών (αναλυτικά βλ. υποενότητα «Αρχικοποίηση μεταβλητών (γραμμή 13)»), ώστε να αποτυπωθεί το αποτέλεσμα «Γινόμενο αρνητικών : 0» από την στιγμή που δεν υπάρχουν αρνητικοί και όχι ψευδώς το αποτέλεσμα «Γινόμενο αρνητικών : 1».

Εκτύπωση του πλήθους των άρτιων αριθμών (γραμμή 70)

Στη γραμμή «70» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «evens» (το πλήθος των άρτιων αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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 μέσου όρου των θετικών αριθμών (γραμμή 71)

Στη γραμμή «71» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «avg» (ο μέσος όρος των θετικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα. Αξίζει να σημειωθεί ότι το αλφαριθμητικό μορφοποίησης είναι το «%20.6f», δηλαδή, το αποτέλεσμα θα τυπωθεί αμεσώς μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του γινομένου των αρνητικών αριθμών (γραμμή 72)

Στη γραμμή «72» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «prod» (το γινόμενο των αρνητικών αριθμών) με μία συνάρτηση «printf()» συνοδευόμενο με το κατάλληλο μήνυμα, κάθε φορά που εκτελείται ο βρόχος. 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.

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1

=====

Μαθηματικές πράξεις με ακέραιους αριθμούς

=====

Εισάγετε ακέραιο αριθμό : 2

Ο ακέραιος αριθμός : [2]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : -3

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Ο ακέραιος αριθμός : [-3]

Τετράγωνο περιττού : [9]

Περιττός αριθμός

Εισάγετε ακέραιο αριθμό : -4

Ο ακέραιος αριθμός : [-4]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : 11

Ο ακέραιος αριθμός : [11]

Τετράγωνο περιττού : [121]

Περιττός αριθμός

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εισάγετε ακέραιο αριθμό : -65

Ο ακέραιος αριθμός : [-65]

Τετράγωνο περιττού : [4225]

Περιττός αριθμός

Εισάγετε ακέραιο αριθμό : 72

Ο ακέραιος αριθμός : [72]

Άρτιος αριθμός

Εισάγετε ακέραιο αριθμό : 0

Ο ακέραιος αριθμός : [0]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

~Τέλος βρόχου~

Πλήθος άρτιων	:	[3]
Μέσος όρος θετικών	:	[28.333334]
Γινόμενο αρνητικών	:	[-780]

Παράδειγμα 2

Μαθηματικές πράξεις με ακέραιους αριθμούς

Εισάγετε ακέραιο αριθμό : 0

Ο ακέραιος αριθμός	:	[0]
--------------------	---	---	----

~Τέλος βρόχου~

Πλήθος άρτιων	:	[0]
Μέσος όρος θετικών	:	[0.000000]
Γινόμενο αρνητικών	:	[0]

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΠΑΡΑΤΗΡΗΣΕΙΣ

Η ορθότητα των αποτελεσμάτων στα παραπάνω παραδείγματα ελέγχθηκε και με τη χρήση τυπικής **αριθμομηχανής** και δεν διαπιστώθηκε καμία διαφορά παρά μόνο σε κάποια λιγότερα σημαντικά ψηφία σε αποτέλεσμα με δεκαδικό μέρος (βλ. «Παράδειγμα 1»), το οποίο δεν επηρεάζει σε μεγάλο βαθμό την ορθότητα των αποτελεσμάτων.

Αναλυτικά, έχουμε δύο παραδείγματα όπου ο χρήστης εισάγει όσους ακέραιους επιθυμεί.

Παράδειγμα 1

Στο παράδειγμα αυτό ο χρήστης εισάγει τους αριθμούς (2, -3, -4, 11, -65, 72). Απο αυτούς οι «3» είναι άρτιοι (2, -4, 72) και οι υπόλοιποι «3» περιττοί (-3, 11, -65). Οι «3» είναι θετικοί (2, 11, 72) και οι υπόλοιποι «3» αρνητικοί (-3, -4, -65).

Ο χρήστης εισάγει στο τέλος και τον αριθμό «0» που σημάνει το τέλος του βρόχου, σαφώς και το τέλος της εισαγωγής ακεραίων αριθμών από τη «standard» είσοδο.

Τετράγωνο περιττού «-3»	:	9	$\{(-3)^2\}$
Τετράγωνο περιττού «11»	:	121	$\{11^2\}$
Τετράγωνο περιττού «-65»	:	4225	$\{(-65)^2\}$
Πλήθος άρτιων	:	3	(2, -4, 72)
Μέσος όρος θετικών	:	28.333334	$\{(2, 11, 72) / 3\} /* (28.333333) */$
Γινόμενο αρνητικών	:	-780	$(-3 * -4 * -65)$

Παράδειγμα 2

Στο παράδειγμα αυτό ο χρήστης εισάγει μόνο τον αριθμό «0» που σημάνει το τέλος του βρόχου, σαφώς και το τέλος της εισαγωγής ακεραίων αριθμών από τη «standard» είσοδο.

Επομένως, δεν υπάρχουν άρτιοι, περιττοί, θετικοί, αρνητικοί αριθμοί. Σαφώς, τα αποτελέσματα είναι τα εξής :

Πλήθος άρτιων	:	0
Μέσος όρος θετικών	:	0.000000
Γινόμενο αρνητικών	:	0

ΘΕΜΑ 4

ΕΠΙΣΗΜΑΝΣΗ «Stars.c»

Το «Πρόγραμμα “Stars.c”» (Πηγαίος Κώδικας) και η «Τεκμηρίωση “Stars.c”» (Ζητούμενο, Δομή, Μεταβλητές, Διάσχιση, Παραδείγματα,

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Παρατηρήσεις) απαντούν στο ζητούμενο του ερωτήματος «Θέμα 4».

ΠΡΟΓΡΑΜΜΑ «Stars.c»

```
1 #include <stdio.h>
2
3 int main (int argc, char **argv)
4
5 {
6     system ("chcp 1253");
7
8     int lines; // Δήλωση μεταβλητών
9     int i, j, k;
10
11     printf ("=====\n\n");
12     printf ("Σχήματα με αστερίσκους\n\n"); // Τίτλος προγράμματος
13     printf ("=====\n\n");
14     printf ("Εισάγετε αριθμό γραμμών : ");
15     scanf ("%d", &lines); // Εισαγωγή του πλήθους των γραμμών
16     printf ("\n-----\n\n");
17     for (i = 1 ; i <= lines ; i++) /* 1ο σχήμα */
18     {
19         for (j = 1 ; j <= i ; j++)
20         {
21             printf ("*");
22         }
23         printf ("\n");
24     }
25     printf ("\n-----\n\n");
26     for (i = 0 ; i < lines ; i++) /* 2ο σχήμα */
27     {
28         for (j = lines - 1 ; j > i ; j--)
29         {
30             printf (" ");
31         }
32         for (k = 0 ; k <= j ; k++)
```


ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
33     {
34         printf ("*");
35     }
36     printf ("\n");
37 }

38     printf      ("\n-----
\n\n");

39 for (i = 0 ; i < lines ; i++) /* 3ο σχήμα */
40 {
41     for (j = lines - 1 ; j > i ; j--)
42     {
43         printf (" ");
44     }
45     for (k = 0 ; k <= j * 2 ; k++)
46     {
47         printf ("*");
48     }
49     printf ("\n");
50 }

51 printf ("\n-----\n\n");

52 for (i = 0 ; i < lines ; i++) /* 4ο σχήμα */
53 {
54     for (j = 0 ; j < lines ; j++)
55     {
56         if (i == 0 || j == 0 || i == lines - 1 || j == lines - 1)
57             /* (~) Σημείο εντός της περιμέτρου του σχήματος */
58             {
59                 printf ("*");
60             }
61             else /* (~) Σημείο εκτός της περιμέτρου του σχήματος */
62             {
63                 if (i == j) /* (!) Σημείο εντός της κύριας διαγωνίου */
64                     printf (".");
65             }
66             else /* (!) Σημείο εκτός της κύριας διαγωνίου */
67             {
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
68                                     if (i + j == lines - 1)
/* (+) Σημείο εντός της δευτερεύουσας διαγωνίου */
69                                     {
70                                         printf (".");
71                                     }
72                                     else /* (+) Σημείο εκτός της δευτερεύουσας
διαγωνίου */
73                                     {
74                                         printf (" ");
75                                     }
76                                     }
77                                 }
78                             }
79                             printf ("\n");
80     }
81
82     return 0;
83 }
```

ΤΕΚΜΗΡΙΩΣΗ «Stars.c»

ΖΗΤΟΥΜΕΝΟ

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

a) Διαβάζει από τη «standard» είσοδο «N» αριθμό γραμμών για την απεικόνιση «4» σχημάτων με αστερίσκους.

b) Απεικονίζει αυτό το σχήμα, για παράδειγμα, για «5» γραμμές :

```
*
**
***
****
*****
```

c) Απεικονίζει αυτό το σχήμα, για παράδειγμα, για «5» γραμμές :

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

**

d) Απεικονίζει αυτό το σχήμα, για παράδειγμα, για «5» γραμμές :

*

e) Απεικονίζει αυτό το σχήμα, για παράδειγμα, για «5» γραμμές :

* . *

* . *

* . *

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 ανάγνωση 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) αριθμητικοί : -, +, *

b) σχεσιακοί : <, >, <=, ==

c) ανάθεσης : =

d) λογικοί : ||

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

e) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης «scanf()» που συνδέεται με τη «standard» είσοδο

f) μετααύξησης : μεταβλητή++

g) μεταμείωσης : μεταβλητή--

Οι εντολές ελέγχου :

a) if – else

Οι εντολές επαναλήψης :

b) for

ΜΕΤΑΒΛΗΤΕΣ

Ακέραιες μεταβλητές (τύπου «int»)

lines (Ο αριθμός των γραμμών των σχημάτων)

i (Μετρητής για τον έλεγχο βρόχου)

j (Μετρητής για τον έλεγχο βρόχου)

k (Μετρητής για τον έλεγχο βρόχου)

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών (γραμμές 8-9)

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στις γραμμές «8-9» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Τίτλος προγράμματος (γραμμή 12)

Στη γραμμή «12» με μία συναρτήση «printf()» τυπώνεται από τη «standard» έξοδο ο τίτλος του προγράμματος (Σχήματα με αστερίσκους), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

Εισαγωγή του πλήθους των γραμμών (γραμμή 15)

Στη γραμμή «15» διαβάζεται από τη «standard» είσοδο με μία συνάρτηση

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

«scanf()» το πλήθος των γραμμών των «4» σχημάτων που θα απεικονιστούν συνοδευόμενο με το κατάλληλο μήνυμα που τυπώνεται από τη «standard» έξοδο με μία συνάρτηση «printf()» (γραμμή 14). Το αποτέλεσμα καταχωρείται στη μεταβλητή «lines» και αξίζει να σημειωθεί ότι τα «4» σχήματα που θα απεικονιστούν, θα έχουν τον ίδιο αριθμό γραμμών που θα εισάγει ο χρήστης (βλ. υποενότητα «Ζητούμενο»).

1^ο σχήμα

(γραμμές 17-24)

Στις γραμμές «17-24» υλοποιείται η σύνταξη για την απεικόνιση του πρώτου σχήματος.

Πιο αναλυτικά, έχουμε έναν βρόχο με μία εντολή επανάληψης «for» (γραμμές 17-24) που εκτελείται για όσο η παράσταση «i <= lines» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής). Μέσα στον βρόχο εκτελείται άλλος ένας βρόχος με μία εντολή επανάληψης «for» (γραμμές 19-22), για όσο η παράσταση «j <= i» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής), και μία συνάρτηση «printf()» (γραμμή 23) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n). Στον δεύτερο βρόχο (γραμμές 19-22) εκτελείται μία συνάρτηση «printf()» (γραμμή 21) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Οι βρόχοι λειτουργούν ως εξής :

a) Πρώτος βρόχος

```
17 for (i = 1 ; i <= lines ; i++)
18 {
19     for (j = 1 ; j <= i ; j++)
20     {
21         printf ("*");
22     }
23     printf ("\n");
24 }
```

Στον πρώτο βρόχο (γραμμές 17-24) έχουμε την μεταβλητή «i» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική της τιμή είναι η τιμή «1» και ο βρόχος εκτελείται μέχρι η «i» που αυξάνεται κατά «1» με την παράσταση «i++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή μεγαλύτερη από το πλήθος των γραμμών των σχημάτων που εισήγαγε ο χρήστης (αναλυτικά βλ. «Εισαγωγή του πλήθους των γραμμών (γραμμή 15)»). Σε κάθε εκτέλεση του βρόχου εκτελείται ο δεύτερος βρόχος (γραμμές 19-22) και η συνάρτηση «printf()» (γραμμή 23) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n). Ο βρόχος εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

b) Δεύτερος βρόχος

```
19 for (j = 1 ; j <= i ; j++)
20 {
21     printf ("*");
22 }
```

Στον δεύτερο βρόχο (γραμμές 19-22) έχουμε την μεταβλητή «j» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική της τιμής είναι η τιμή «1» και ο βρόχος εκτελείται μέχρι η «j» που αυξάνεται κατά «1» με την παράσταση «j++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή μεγαλύτερη από την βοηθητική «i» του πρώτου βρόχου (γραμμές 17-24).

Ο βρόχος εκτελείται κάθε φορά που εκτελείται ο πρώτος βρόχος και οι φορές που εκτελούνται αυξάνεται κατά «1», καθώς από τον πρώτο βρόχο η βοηθητική «i» αυξάνεται κατά «1» με την παράσταση «i++». Σε κάθε εκτέλεση του βρόχου εκτελείται η συνάρτηση «printf()» (γραμμή 21) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Για παράδειγμα, ο χρήστης εισάγει «5» γραμμές που καταχωρείται στη μεταβλητή «lines». Άρα, ο πρώτος βρόχος εκτελείται «5» φορές, ενώ, ο δεύτερος εκτελείται :

- a) Στην 1^η επανάληψη του πρώτου βρόχου, «1» φορά (*\n).
- b) Στην 2^η επανάληψη του πρώτου βρόχου, «2» φορές (**\n).
- c) Στην 3^η επανάληψη του πρώτου βρόχου, «3» φορές (***\n).
- d) Στην 4^η επανάληψη του πρώτου βρόχου, «4» φορές (****\n).
- e) Στην 5^η επανάληψη του πρώτου βρόχου, «5» φορές (*****\n).

Το σχήμα απεικονίζεται ως εξής :

```
*
**
***
****
*****
```

2^ο σχήμα

(γραμμές 26-37)

Στις γραμμές «26-37» υλοποιείται η σύνταξη για την απεικόνιση του

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

δεύτερου σχήματος.

Πιο αναλυτικά, έχουμε έναν βρόχο με μία εντολή επανάληψης «for» (γραμμές 26-37) που εκτελείται για όσο η παράσταση «i < lines» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής). Μέσα στον βρόχο εκτελούνται άλλοι δύο βρόχοι με την εντολή επανάληψης «for». Ο ένας για όσο η παράσταση «j > i» (γραμμές 28-31) παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής) και ο άλλος για όσο η παράσταση «k <= j» (γραμμές 32-35) παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής). Επίσης, εκτελείται μία συνάρτηση «printf()» (γραμμή 36) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n). Στον δεύτερο βρόχο (γραμμές 28-31) εκτελείται μία συνάρτηση «printf()» (γραμμή 30) που τυπώνει τον χαρακτήρα κενό (" "), ενώ στον τρίτο εκτελείται, επίσης, μία συνάρτηση «printf()» (γραμμή 34) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Οι βρόχοι λειτουργούν ως εξής :

a) Πρώτος βρόχος

```
26 for (i = 0 ; i < lines ; i++)
27 {
28     for (j = lines - 1 ; j > i ; j--)
29     {
30         printf (" ");
31     }
32     for (k = 0 ; k <= j ; k++)
33     {
34         printf ("*");
35     }
36     printf ("\n");
37 }
```

Στον πρώτο βρόχο (γραμμές 26-37) έχουμε την μεταβλητή «i» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι η βοηθητική, που αυξάνεται κατά «1» με την παράσταση «i++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή μεγαλύτερη από το πλήθος των γραμμών των σχημάτων που εισήγαγε ο χρήστης (γραμμή 15). Σε κάθε εκτέλεση του βρόχου εκτελείται ο δεύτερος βρόχος (γραμμές 28-31), ο τρίτος (γραμμές 32-35) και μία συνάρτηση «printf()» (γραμμή 36) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n).

Ο βρόχος εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο.

b) Δεύτερος βρόχος

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
28         for (j = lines - 1 ; j > i ; j--)
29         {
30             printf (" ");
31         }
```

Στον δεύτερο βρόχο (γραμμές 28-31) έχουμε την μεταβλητή «j» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «lines - 1», δηλαδή, ο αριθμός των γραμμών που εισάγει ο χρήστης κατά μείον «1» και ο βρόχος εκτελείται μέχρι η βοηθητική, που μειώνεται κατά «1» με την παράσταση «j--» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή μικρότερη από την βοηθητική «i» του πρώτου βρόχου που αυξάνεται κατά «1» με την παράσταση «i++».

Ο βρόχος στην αρχή εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο μειωμένες κατά «1» (γραμμή 15) και φυσικά, όταν εκτελείται ο πρώτος βρόχος (γραμμές 26-37). Οι φορές που εκτελείται μειώνονται κατά «1», καθώς από τον δεύτερο βρόχο η βοηθητική «j» μειώνεται κατά «1» με την παράσταση «j--». Σε κάθε εκτέλεση του βρόχου εκτελείται η συνάρτηση «printf()» (γραμμή 30) που τυπώνει τον χαρακτήρα κενό (" ").

c) Τρίτος βρόχος

```
32 for (k = 0 ; k <= j ; k++)
33 {
34     printf ("*");
35 }
```

Στον τρίτο βρόχο (γραμμές 32-35) έχουμε την μεταβλητή «k» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι η βοηθητική που αυξάνεται κατά «1» με την παράσταση «k++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή μεγαλύτερη από την βοηθητική «j» του δεύτερου βρόχου που μειώνεται κατά «1» με την παράσταση «j--».

Ο βρόχος εκτελείται κάθε φορά που εκτελείται ο πρώτος βρόχος (γραμμές 26-37) και οι φορές που εκτελούνται αυξάνονται κατά «1», καθώς από τον πρώτο βρόχο η βοηθητική «i» αυξάνεται κατά «1» με την παράσταση «i++» και αφού εκτελεστεί ο δεύτερος βρόχος, ώστε να μειωθεί η βοηθητική «j» κατά «1» με την παράσταση «j--» που βρίσκεται στην παράσταση «k <= j» που ελέγχει τον τρίτο βρόχο. Σε κάθε εκτέλεση του βρόχου εκτελείται η συνάρτηση «printf()» (γραμμή 34) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Για παράδειγμα, ο χρήστης εισάγει «5» γραμμές που καταχωρείται στη μεταβλητή «lines». Άρα, ο πρώτος βρόχος εκτελείται «5» φορές, ενώ, ο δεύτερος και ο τρίτος εκτελούνται :

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

a) Στην 1^η επανάληψη του πρώτου βρόχου εκτελείται :

«4» φορές ο 2^{ος} βρόχος, «1» φορά ο 3^{ος} βρόχος (* \backslash n).

b) Στην 2^η επανάληψη του πρώτου βρόχου εκτελείται :

«3» φορές ο 2^{ος} βρόχος, «2» φορές ο 3^{ος} βρόχος (** \backslash n).

c) Στην 3^η επανάληψη του πρώτου βρόχου εκτελείται :

«2» φορές ο 2^{ος} βρόχος, «3» φορές ο 3^{ος} βρόχος (*** \backslash n).

d) Στην 4^η επανάληψη του πρώτου βρόχου εκτελείται :

«1» φορά ο 2^{ος} βρόχος, «4» φορές ο 3^{ος} βρόχος (**** \backslash n).

e) Στην 5^η επανάληψη του πρώτου βρόχου εκτελείται :

«0» φορές ο 2^{ος} βρόχος, «5» φορές ο 3^{ος} βρόχος (***** \backslash n).

Το σχήμα απεικονίζεται ως εξής :

```
*  
**  
***  
****  
*****
```

3^ο σχήμα

(γραμμές 39-50)

Στις γραμμές «39-50» υλοποιείται η σύνταξη για την απεικόνιση του τρίτου σχήματος.

Πιο αναλυτικά, έχουμε έναν βρόχο με μία εντολή επανάληψης «for» (γραμμές 39-50) που εκτελείται για όσο η παράσταση «i < lines» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθείας). Μέσα στο βρόχο εκτελούνται άλλοι δύο βρόχοι με την εντολή επανάληψης «for». Ο ένας για όσο η παράσταση «j > i» (γραμμές 41-44) παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθείας) και ο άλλος για όσο η παράσταση «k <= j * 2» (γραμμές 45-48) παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθείας). Επίσης, εκτελείται μία συνάρτηση «printf()» (γραμμή 49) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\backslash n). Στον δεύτερο βρόχο (γραμμές 41-44) εκτελείται μία συνάρτηση «printf()» (γραμμή 43) που τυπώνει τον χαρακτήρα κενό (" "), ενώ στον τρίτο εκτελείται, επίσης, μία συνάρτηση «printf()» (γραμμή 47) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Οι βρόχοι λειτουργούν ως εξής :

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

a) Πρώτος βρόχος

```
39 for (i = 0 ; i < lines ; i++)
40 {
41     for (j = lines - 1 ; j > i ; j--)
42     {
43         printf (" ");
44     }
45     for (k = 0 ; k <= j * 2 ; k++)
46     {
47         printf ("*");
48     }
49     printf ("\n");
50 }
```

Στον πρώτο βρόχο (γραμμές 39-50) έχουμε τη μεταβλητή «i» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι η βοηθητική «i», που αυξάνεται κατά «1» με την παράσταση «i++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή με γαλύτερη από το πλήθος των γραμμών των σχημάτων που εισήγαγε ο χρήστης (γραμμή 15). Σε κάθε εκτέλεση του βρόχου εκτελείται ο δεύτερος βρόχος (γραμμές 41-44), ο τρίτος (γραμμές 45-48) και μία συνάρτηση «printf()» (γραμμή 49) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n).

Ο βρόχος εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο.

b) Δεύτερος βρόχος

```
41     for (j = lines - 1 ; j > i ; j--)
42     {
43         printf (" ");
44     }
```

Στον δεύτερο βρόχο (γραμμές 41-44) έχουμε τη μεταβλητή «j» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «lines - 1», δηλαδή, ο αριθμός των γραμμών που εισάγει ο χρήστης κατά μείον «1» και ο βρόχος εκτελείται μέχρι ο μετρητής που μειώνεται κατά «1» με την παράσταση «j--» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή μικρότερη από την βοηθητική «i» του πρώτου βρόχου που αυξάνεται κατά «1» με την παράσταση «i++» μετά από κάθε εκτέλεση του.

Ο βρόχος στην αρχή εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο μειωμένες

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

κατά «1» (γραμμή 15) και φυσικά, όταν εκτελείται ο πρώτος βρόχος (γραμμές 39-50). Οι φορές που εκτελείται μειώνονται κατά «1», καθώς από τον πρώτο βρόχο ο μετρητής «j» μειώνεται κατά «1» με την παράσταση «j--». Σε κάθε εκτέλεση του βρόχου εκτελείται η συνάρτηση «printf()» (γραμμή 43) που τυπώνει τον χαρακτήρα κενό (" ").

c) Τρίτος βρόχος

```
45 for (k = 0 ; k <= j * 2 ; k++)
46     {
47         printf ("*");
48     }
```

Στον τρίτο βρόχο (γραμμές 45-48) έχουμε τη μεταβλητή «k» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι ο μετρητής που αυξάνεται κατά «1» με την παράσταση «k++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή μεγαλύτερη από τη παράσταση «j * 2» με όρο την βοηθητική «j» του δεύτερου βρόχου που μειώνεται κατά «1» με την παράσταση «j--».

Ο βρόχος εκτελείται κάθε φορά που εκτελείται ο πρώτος βρόχος (γραμμές 39-50) και οι φορές που εκτελούνται αυξάνεται κατά «2», καθώς από τον πρώτο βρόχο η βοηθητική «i» αυξάνεται κατά «1» με την παράσταση «i++» και αφού εκτελεστεί ο δεύτερος βρόχος, ώστε να μειωθεί η βοηθητική «j» κατά «1» με την παράσταση «j--» που βρίσκεται στην παράσταση «k <= j * 2» που ελέγχει τον τρίτο βρόχο. Σε κάθε εκτέλεση του βρόχου εκτελείται η συνάρτηση «printf()» (γραμμή 47) που τυπώνει τον χαρακτήρα αστερίσκο (*).

Για παράδειγμα, ο χρήστης εισάγει «5» γραμμές που καταχωρείται στη μεταβλητή «lines». Άρα, ο πρώτος βρόχος εκτελείται «5» φορές, ενώ, ο δεύτερος και ο τρίτος εκτελούνται :

a) Στην 1^η επανάληψη του πρώτου βρόχου εκτελείται :

«4» φορές ο 2^{ος} βρόχος, «1» φορά ο 3^{ος} βρόχος (*
*\n).

b) Στην 2^η επανάληψη του πρώτου βρόχου εκτελείται :

«3» φορές ο 2^{ος} βρόχος, «3» φορές ο 3^{ος} βρόχος (***\n).

c) Στην 3^η επανάληψη του πρώτου βρόχου εκτελείται :

«2» φορές ο 2^{ος} βρόχος, «5» φορές ο 3^{ος} βρόχος (*****\n).

d) Στην 4^η επανάληψη του πρώτου βρόχου εκτελείται :

«1» φορά ο 2^{ος} βρόχος, «7» φορές ο 3^{ος} βρόχος (*****\n).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ε) Στην 5^η επανάληψη του πρώτου βρόχου εκτελείται :

«0» φορές ο 2^{ος} βρόχος, «9» φορές ο 3^{ος} βρόχος (*****\n).

Το σχήμα απεικονίζεται ως εξής :

```
*  
  
***  
  
*****  
  
*****  
  
*****
```

4^ο σχήμα

(γραμμές 52-80)

Στις γραμμές «52-80» υλοποιείται η σύνταξη για την απεικόνιση του τέταρτου σχήματος.

Πιο αναλυτικά, έχουμε έναν βρόχο με μία εντολή επανάληψης «for» (γραμμές 52-80) που εκτελείται για όσο η παράσταση «i < lines» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής). Μέσα στο βρόχο εκτελείται άλλος ένας βρόχος με μία εντολή επανάληψης «for» (γραμμές 54-78), για όσο η παράσταση «j < lines» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής), μία συνάρτηση «printf()» (γραμμή 79) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n) και μία εντολή ελέγχου «if-else» (γραμμή 56-76).

Οι βρόχοι λειτουργούν ως εξής :

a) Πρώτος βρόχος

```
52 for (i = 0 ; i < lines ; i++)  
53 {  
54     for (j = 0 ; j < lines ; j++)  
55     {  
56         if (i == 0 || j == 0 || i == lines - 1 || j == lines - 1)  
            /* Σημείο εντός της περιμέτρου του σχήματος */  
57         {  
58             printf ("*");  
59         }  
60         else // Σημείο εκτός της περιμέτρου του σχήματος  
61         {  
62             if (i==j)  
                /* Σημείο εντός της κύριας διαγωνίου */  
63             {  
64                 printf (".");  
65             }
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
66             else
67             /* Σημείο εκτός της κύριας διαγωνίου */
68             {
69                 if (i + j == lines - 1)
70                 /* Σημείο εντός της δευτερεύουσας διαγωνίου */
71                 {
72                     printf (".");
73                 }
74             else
75             /* Σημείο εκτός της δευτερεύουσας διαγωνίου */
76             {
77                 printf (" ");
78             }
79         printf ("\n");
80     }
```

Στον πρώτο βρόχο (γραμμές 52-80) έχουμε τη μεταβλητή «i» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι η βοηθητική που αυξάνεται κατά «1» με την παράσταση «i++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή μεγαλύτερη από το πλήθος των γραμμών των σχημάτων που εισήγαγε ο χρήστης (γραμμή 15). Σε κάθε εκτέλεση του βρόχου εκτελείται ο δεύτερος βρόχος (γραμμές 54-78) και η συνάρτηση «printf()» (γραμμή 79) που τυπώνει τον χαρακτήρα διαφυγής της αλλαγής γραμμής (\n).

Ο βρόχος εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο.

b) Δεύτερος βρόχος

```
54     for (j = 0 ; j < lines ; j++)
55     {
56         if (i == 0 || j == 0 || i == lines - 1 || j == lines - 1)
57         /* Σημείο εντός της περιμέτρου του σχήματος */
58         {
59             printf ("*");
60         }
61         else // Σημείο εκτός της περιμέτρου του σχήματος
62         {
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
62             if (i==j)
/* Σημείο εντός της κύριας διαγωνίου */
63             {
64                 printf (".");
65             }
66             else
/* Σημείο εκτός της κύριας διαγωνίου */
67             {
68                 if (i + j == lines - 1)
/* Σημείο εντός της δευτερεύουσας διαγωνίου */
69                 {
70                     printf (".");
71                 }
72                 else
/* Σημείο εκτός της δευτερεύουσας διαγωνίου */
73                 {
74                     printf (" ");
75                 }
76             }
77         }
78     }
```

Στον δεύτερο βρόχο (γραμμές 54-78) έχουμε τη μεταβλητή «j» (τύπου «int») που χρησιμοποιείται ως βοηθητική για τον έλεγχο του.

Η αρχική του τιμή είναι η τιμή «0» και ο βρόχος εκτελείται μέχρι ο μετρητής που αυξάνεται κατά «1» με την παράσταση «j++» κάθε φορά που τελειώνει μία εκτέλεση του βρόχου, περιέχει τιμή ίση ή μεγαλύτερη από το πλήθος των γραμμών των σχημάτων που εισήγαγε ο χρήστης (γραμμή 15). Σε κάθε εκτέλεση του βρόχου εκτελείται η εντολή «if-else» που ελέγχει τη θέση ενός σημείου στον τετραγωνικό πίνακα που είναι το τέτατρο σχήμα με συντεταγμένες [i,j], ώστε να τυπώσει τους κατάλληλους χαρακτήρες (αναλυτικά βλ. «(~) Σημείο εντός της περιμέτρου του σχήματος (γραμμές 56-59)», «Σημείο εκτός της περιμέτρου του σχήματος (γραμμές 60-77)», «Σημείο εντός της κύριας διαγωνίου (γραμμές 62-65)», «Σημείο εκτός της κύριας διαγωνίου (γραμμές 66-76)», «Σημείο εντός της δευτερεύουσας διαγωνίου (γραμμές 68-71)», «Σημείο εκτός της δευτερεύουσας διαγωνίου (γραμμές 72-75)»).

Ο βρόχος εκτελείται τόσες φορές όσες είναι οι γραμμές των σχημάτων που εισήγαγε ο χρήστης από τη «standard» είσοδο κάθε φορά που εκτελείται ο πρώτος βρόχος (γραμμές 52-80). Επομένως, εκτελείται «lines * lines» φορές. ***(Σελίδα 59)**

(~) Σημείο εντός της περιμέτρου του σχήματος	(γραμμές 56-59)
(~) Σημείο εκτός της περιμέτρου του σχήματος	(γραμμές 60-77)

Στις γραμμές «56-77» υλοποιείται με μία εντολή ελέγχου «if-else» ένας

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

έλεγχος για την περίπτωση που το σημείο με συντεταγμένες $[i,j]$ βρίσκεται εντός ή εκτός της περιμέτρου του σχήματος του τετραγωνικού πίνακα.

Διεξοδικά, αν η παράσταση « $i == 0 \parallel j == 0 \parallel i == \text{lines} - 1 \parallel j == \text{lines} - 1$ » παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστούν οι εντολές της υποενότητας «(~) Σημείο εντός της περιμέτρου του σχήματος» στις γραμμές «56-59», διαφορετικά αν παράγει αποτέλεσμα την τιμή «False», τότε θα εκτελεστούν οι εντολές της υποενότητας «(~) Σημείο εκτός της παραμέτρου του σχήματος» στις γραμμές «60-77».

(~) Σημείο εντός της περιμέτρου του σχήματος (γραμμές 56-59)

Στις γραμμές «56-59» εκτελούνται οι εντολές της «if ($i == 0 \parallel j == 0 \parallel i == \text{lines} - 1 \parallel j == \text{lines} - 1$)» (γραμμή 56) για την περίπτωση που το σημείο με συντεταγμένες $[i,j]$ βρίσκεται εντός της περιμέτρου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση « $i == 0 \parallel j == 0 \parallel i == \text{lines} - 1 \parallel j == \text{lines} - 1$ » παράγει μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστεί η συνάρτηση «printf()» (γραμμή 58) που τυπώνει τον χαρακτήρα αστερίσκο (*). Πρόκειται για μία λογική παράσταση με τον λογικό τελεστή «||» (OR), οπότε θα παράγει αποτέλεσμα μία τιμή «True» αν μία τουλάχιστον από τις παραστάσεις « $i == 0$, $j == 0$, $i == \text{lines} - 1$, $j == \text{lines} - 1$ » παράγουν αποτέλεσμα μία τιμή «True».

Για παράδειγμα, σημεία που ανήκουν στην περίμετρο ενός τετραγωνικού πίνακα διαστάσεων «5X5» είναι τα σημεία με συντεταγμένες [0.0], [0.1], [0.2], [0.3], [0.4], [1.4], [2.4], [3.4], [4.4], [4.3], [4.2], [4.1], [4.0], [3.0], [2.0], [1.0].

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

(~) Σημείο εκτός της περιμέτρου του σχήματος (γραμμές 60-77)

Στις γραμμές «60-77» εκτελούνται οι εντολές της «else» (γραμμή 60) που αντιστοιχεί στην «if ($i == 0 \parallel j == 0 \parallel i == \text{lines} - 1 \parallel j == \text{lines} - 1$)» (γραμμή 56) για την περίπτωση που το σημείο με συντεταγμένες $[i,j]$ βρίσκεται εκτός της περιμέτρου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση « $i == 0 \parallel j == 0 \parallel i == \text{lines} - 1 \parallel j == \text{lines} - 1$ » παράγει μία τιμή «False» (συνθήκη ψευδής), τότε θα εκτελεστεί η εντολή «if-else» (γραμμές 62-76) που ελέγχει αν το σημείο με συντεταγμένες $[i,j]$ ανήκει σε κύρια ή δευτερεύουσα διαγώνιο. Πρόκειται για μία λογική παράσταση με τον λογικό τελεστή «||» (OR), οπότε θα παράγει αποτέλεσμα μία τιμή «False», όταν, όλες οι παραστάσεις « $i == 0$, $j == 0$, $i == \text{lines} - 1$, j

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

`== lines - 1` παράγουν αποτέλεσμα μία τιμή «False».

Για παράδειγμα, σημεία που δεν ανήκουν στην περίμετρο ενός τετραγωνικού πίνακα διαστάσεων «5X5» είναι τα σημεία με συντεταγμένες [1.1], [1.2], [1.3], [2.1], [2.2], [2.3], [3.1], [3.2], [3.3].

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

(!) Σημείο εντός της κυρίας διαγωνίου

(γραμμές 62-65)

(!) Σημείο εκτός της κυρίας διαγωνίου

(γραμμές 66-76)

Στις γραμμές «62-76» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εντός ή εκτός της κυρίας διαγωνίου και ταυτόχρονα εκτός της περιμέτρου του σχήματος του τετραγωνικού πίνακα.

Διεξοδικά, αν η παράσταση «`i == j`» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστούν οι εντολές της υποενότητας «(!) Σημείο εντός της κυρίας διαγωνίου» στις γραμμές «62-65», διαφορετικά αν παράγει αποτέλεσμα μία τιμή «False», τότε θα εκτελεστούν οι εντολές της υποενότητας «(!) Σημείο εκτός της κυρίας διαγωνίου» στις γραμμές «66-76».

(!) Σημείο εντός της κυρίας διαγωνίου

(γραμμές 62-65)

Στις γραμμές «62-65» εκτελούνται οι εντολές της «if (i == j)» (γραμμή 62) για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εντός της κυρίας διαγωνίου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση «`i == j`» παράγει μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστεί η συνάρτηση «`printf()`» (γραμμή 64) που τυπώνει τον χαρακτήρα τελεία (.). Δηλαδή, όταν, οι μεταβλητές «i» και «j» περιέχουν την ίδια τιμή.

Για παράδειγμα, σημεία που ανήκουν στην κύρια διαγώνιο και εκτός της περιμέτρου ενός τετραγωνικού πίνακα διαστάσεων «5X5» είναι τα σημεία με συντεταγμένες [1.1], [2.2], [3.3].

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

(!) Σημείο εκτός της κυρίας διαγωνίου (γραμμές 66-76)

Στις γραμμές «66-76» εκτελούνται οι εντολές της «else» (γραμμή 66) που αντιστοιχεί στην «if (i == j)» (γραμμή 62) για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εκτός της κυρίας διαγωνίου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση «i == j» παράγει μία τιμή «False» (συνθήκη αληθής), τότε θα εκτελεστεί η εντολή «if-else» (γραμμές 68-75) που ελέγχει αν το σημείο με συντεταγμένες [i,j] ανήκει στην δευτερεύουσα διαγώνιο ή όχι. Δηλαδή, όταν, οι μεταβλητές «i» και «j» δεν περιέχουν την ίδια τιμή.

Για παράδειγμα, σημεία που δεν ανήκουν στην κύρια διαγώνιο και εκτός περιμέτρου ενός τετραγωνικού πίνακα είναι τα σημεία με συντεταγμένες [0.0], [0.1], [0.2], [0.3], [0.4], [1.0], [1.2], [1.3], [1.4], [2.0], [2.1], [2.3], [2.4], [3.0], [3.1], [3.2], [3.4], [4.0], [4.1], [4.2], [4.3], [4.4].

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

(+) Σημείο εντός της δευτερεύουσας διαγωνίου (γραμμές 68-71)
(+) Σημείο εκτός της δευτερεύουσας διαγωνίου (γραμμές 72-75)

Στις γραμμές «68-75» υλοποιείται με μία εντολή ελέγχου «if-else» ένας έλεγχος για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εντός ή εκτός της δευτερεύουσας διαγωνίου και ταυτόχρονα εκτός της περιμέτρου του σχήματος του τετραγωνικού πίνακα.

Διεξοδικά, αν η παράσταση «i + j == lines - 1» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστούν οι εντολές της υποενότητας «(+) Σημείο εντός της δευτερεύουσας διαγωνίου» στις γραμμές «68-71», διαφορετικά, αν παράγει αποτέλεσμα μία τιμή «False»,

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

τότε θα εκτελεστούν οι εντολές της υποενότητας «(+) Σημείο εκτός της δευτερεύουσας διαγωνίου» στις γραμμές «72-75».

(+) Σημείο εντός της δευτερεύουσας διαγωνίου (γραμμές 68-71)

Στις γραμμές «68-71» εκτελούνται οι εντολές της «if (i + j == lines - 1)» (γραμμή 68) για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εντός της δευτερεύουσας διαγωνίου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση «i + j == lines - 1» παράγει μία τιμή «True» (συνθήκη αληθής), τότε θα εκτελεστεί η συνάρτηση «printf()» (γραμμή 64) που τυπώνει τον χαρακτήρα τελεία (.). Δηλαδή, όταν, το άθροισμα των μεταβλητών «i» και «j» περιέχουν την ίδια τιμή με τον αριθμό των γραμμών (lines) μειομένο κατά «1».

Για παράδειγμα, σημεία που ανήκουν στην δευτερεύουσα διαγώνιο και εκτός της περιμέτρου ενός τετραγωνικού πίνακα διαστάσεων «5X5» είναι τα σημεία με συντεταγμένες [1.3], [2.2], [3.1].

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

(+) Σημείο εκτός της δευτερεύουσας διαγωνίου (γραμμές 72-75)

Στις γραμμές «72-75» εκτελούνται οι εντολές της «else» (γραμμή 72) που αντιστοιχεί στην «if (i + j == lines - 1)» (γραμμή 68) για την περίπτωση που το σημείο με συντεταγμένες [i,j] βρίσκεται εκτός της δευτερεύουσας διαγωνίου του σχήματος του τετραγωνικού πίνακα.

Πιο αναλυτικά, αν η παράσταση «i + j == lines - 1» παράγει μία τιμή «False» (συνθήκη αληθής), τότε θα εκτελεστεί η συνάρτηση «printf()» (γραμμή 74) που τυπώνει τον χαρακτήρα κενό (" "). Δηλαδή, όταν, το άθροισμα των μεταβλητών «i» και «j» δεν περιέχουν περιέχουν την ίδια τιμή με τον αριθμό των γραμμών (lines) μειομένο κατά «1».

Για παράδειγμα, σημεία που δεν ανήκουν στην δευτερεύουσα διαγώνιο και εκτός της περιμέτρου ενός τετραγωνικού πίνακα είναι τα σημεία με συντεταγμένες [0.0], [0.2], [0.3], [0.4], [1.0], [1.1], [1.2], [1.4], [2.0], [2.1], [2.3], [2.4], [3.0], [3.2], [3.3], [3.4], [4.0], [4.1], [4.2], [4.3], [4.4].

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

	0	1	2	3	4
0	[0.0]	[0.1]	[0.2]	[0.3]	[0.4]
1	[1.0]	[1.1]	[1.2]	[1.3]	[1.4]
2	[2.0]	[2.1]	[2.2]	[2.3]	[2.4]
3	[3.0]	[3.1]	[3.2]	[3.3]	[3.4]
4	[4.0]	[4.1]	[4.2]	[4.3]	[4.4]

* Για παράδειγμα, ο χρήστης εισάγει «5» γραμμές που καταχωρείται στη μεταβλητή «lines». Άρα, ο πρώτος βρόχος εκτελείται «5» φορές, ενώ, ο δεύτερος εκτελείται :

- a) Στην 1^η επανάληψη του πρώτου βρόχου, «5» φορές.
- b) Στην 2^η επανάληψη του πρώτου βρόχου, «5» φορές.
- c) Στην 3^η επανάληψη του πρώτου βρόχου, «5» φορές.
- d) Στην 4^η επανάληψη του πρώτου βρόχου, «5» φορές.
- e) Στην 5^η επανάληψη του πρώτου βρόχου, «5» φορές.

Το σχήμα απεικονίζεται ως εξής :

```
*****
*   *
*   *
*   *
*   *
*   *
*****
```

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1 (8 γραμμές)

=====

Σχήματα με αστερίσκους

=====

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εισάγετε αριθμό γραμμών : 8

```
*  
  
**  
  
***  
  
****  
  
*****  
  
*****  
  
*****  
  
*****
```

```
  *  
  
  **  
  
  ***  
  
  ****  
  
  *****  
  
  *****  
  
  *****  
  
  *****
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

*

* . *

* . *

* .. *

* .. *

* . *

* . *

Παράδειγμα 2 (2 γραμμές)

Active code page: 1253

=====

Σχήματα με αστερίσκους

=====

Εισάγετε αριθμό γραμμών : 2

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

*

**

*

**

*

**

**

ΠΑΡΑΤΗΡΗΣΕΙΣ

Αναλυτικά, έχουμε δύο παραδείγματα όπου ο χρήστης εισάγει «8» γραμμές και «2» για να αναπαραστήσει τα σχήματα.

Παράδειγμα 1 (5 γραμμές)

Τα τέσσερα σχήματα απεικονίζονται με 5 γραμμές.

Παράδειγμα 2 (2 γραμμές)

Τα τέσσερα σχήματα απεικονίζονται με 2 γραμμές. Αξίζει να σημειωθεί ότι στο τέταρτο σχήμα δεν υπάρχει σημείο εκτός της περιμέτρου του σχήματος και εντός της κύριας ή δευτερεύουσας διαγωνίου ταυτόχρονα, γι' αυτό και δεν τυπώνεται ο χαρακτήρας τελεία (.).

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΘΕΜΑ 5

ΕΠΙΣΗΜΑΝΣΗ «SinTaylor.c»

Το «Πρόγραμμα “SinTaylor.c”» (Πηγαίος Κώδικας) και η «Τεκμηρίωση “SinTaylor.c”» (Ζητούμενο, Δομή, Μεταβλητές, Διάσχιση, Παραδείγματα, Παρατηρήσεις) απαντούν στο ζητούμενο του ερωτήματος «Θέμα 5».

ΠΡΟΓΡΑΜΜΑ «SinTaylor.c»

```
1 #include <stdio.h>
2 #include <math.h>
3 #define pi 3.14159
4
5 int main (int argc, char **argv)
6 {
7     system ("chcp 1253");
8
9     double deg, rad; // Δήλωση μεταβλητών
10    double sine;
11    double term, next_term, diff_terms, abs_diff_terms, first_sin_T;
12    double diff_sin_T, abs_diff_sin_T;
13    double sin_T = 0.0; // Αρχικοποίηση μεταβλητών
14    int i;
15    int j = 1; // Αρχικοποίηση μεταβλητών
16    int sign = -1; // Αρχικοποίηση μεταβλητών
17
18    printf ("=====\n\n");
19    printf ("Υπολογισμός του ημιτόνου μίας γωνίας\n\n"); // Ο τίτλος του προγράμματος
20    printf ("=====\n\n");
21    printf ("Εισάγετε γωνία στο διάστημα του 1ου κύκλου [0,360]\n\n");
22    printf ("Μοίρες : ");
23    scanf ("%lf", &deg); // Εισαγωγή της γωνίας σε μοίρες
24    printf ("\n-----\n\n");
```

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
25  rad = (pi * deg) / 180; // Μετατροπή της γωνίας από μοίρες σε ακτίνια
26  printf ("Μοίρες : [%20.6lf]\n", deg); // Εκτύπωση της γωνίας σε μοίρες
27  printf ("Ακτίνια : [%20.6lf]\n\n", rad); // Εκτύπωση της γωνίας σε
    ακτίνια
28  sine = sin (rad); // Υπολογισμός του ημιτόνου της γωνίας με τη συνάρτηση
    "sin (ω)"
29  do /* 1ος Βρόχος */
30  {
31      term = 1; // Αρχικοποίηση μεταβλητής του πρώτου όρου, του
    δευτέρου ...
32      for (i = 1 ; i <= j ; i++) /* 2ος Βρόχος */
33      {
34          term = term * (rad / i); // Υπολογισμός του πρώτου όρου,
    του δευτέρου ... ( $\omega^1 / 1!$ ,  $\omega^3 / 3!$  ...)
35      }
36      j = j + 2; // Αύξηση της βοηθητικής μεταβλητής για τον υπολογισμό
    του δευτέρου όρου, του τρίτου ...
37      next_term = term * ((rad * rad) / (j * (j - 1)));
    // Υπολογισμός του δευτέρου όρου, του τρίτου ... ( $\omega^3 / 3!$ ,  $\omega^5 / 5!$  ...)
38      diff_terms = next_term - term; // Υπολογισμός της διαφοράς του
    δευτέρου όρου με τον πρώτο, του τρίτου με τον δεύτερο ...
39      abs_diff_terms = fabs (diff_terms); // Υπολογισμός της απόλυτης
    τιμής της διαφοράς
40      if (j == 3) /* (~) 1η επανάληψη του 1ου βρόχου */
41      { /*  $\omega^1 / 1! - \omega^3 / 3!$  */
42          first_sin_T = term + (sign * next_term); // Υπολογισμός
    του πρώτου αθροίσματος με το κατάλληλο πρόσημο "sign"
43          sin_T = sin_T + first_sin_T; // Καταχώρηση του
    αθροίσματος στη μεταβλητή "sin_t"
44          sign = sign * (-1); // Αλλαγή προσήμου
45      }
46      else /* (~) 2η, 3η ... επανάληψη του 1ου βρόχου */
47      { /* ( $\omega^1 / 1! - \omega^3 / 3!$ ) +  $\omega^5 / 5!$  ... */
48          sin_T = sin_T + (sign * next_term); // Υπολογισμός του
    δευτέρου αθροίσματος, του τρίτου ... με το κατάλληλο πρόσημο "sign"
49          sign = sign * (-1); // Αλλαγή προσήμου
50      }
51  }
52  while (abs_diff_terms > 0.000001);
53
54  printf ("Sine : [%20.6lf]\n", sine); // Εκτύπωση του ημιτόνου της
    γωνίας με την έτοιμη συνάρτηση "sin (ω)"
```


ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

```
55 printf ("Taylor : [%20.6lf]\n\n", sin_T); // Εκτύπωση του ημιτόνου
της γωνίας με την απειροσειρά "Taylor"

56 diff_sin_T = sine - sin_T; // Υπολογισμός της διαφοράς του ημιτόνου με
την απειροσειρά "Taylor"

57 abs_diff_sin_T = fabs (diff_sin_T); // Υπολογισμός της απόλυτης τιμής
της διαφοράς του ημιτόνου με την απειροσειρά "Taylor"

58 if (abs_diff_sin_T <= 0.0000009) /* (~) Αποδεκτή απόλυτη τιμή της
διαφοράς του ημιτόνου με την απειροσειρά "Taylor" */

59 {

60     printf ("Sine ~= Taylor\n");
61     printf ("Οι δύο αριθμοί είναι σχεδόν ίσοι\n");
62 }

63 else /* (~) Απορριπτέα απόλυτη τιμή της διαφοράς του ημιτόνου με την
απειροσειρά "Taylor" */

64 {

65     printf ("Sine != Taylor\n");
66     printf ("Οι δύο αριθμοί δεν είναι σχεδόν ίσοι\n");
67 }

68

69 return 0;

70 }
```

ΤΕΚΜΗΡΙΩΣΗ «SinTaylor.c»

ΖΗΤΟΥΜΕΝΟ

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

- a) Διαβάζει από τη «standard» είσοδο μία γωνία σε μοίρες.
- b) Τη μετατρέπει σε ακτίνια (ω).
- c) Υπολογίζει το ημίτονο της γωνίας σε ακτίνια με την έτοιμη συνάρτηση «sin (ω)» της βιβλιοθήκης «math.h».
- d) Υπολογίζει το ημίτονο της γωνίας σε ακτίνια με την χαρακτηριστική απειροσειρά «Taylor».

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \dots$$

- e) Συγκρίνει τους δύο αριθμούς και καταλήγει στο πόρισμα αν είναι «σχεδόν»

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ίσοι ή όχι.

f) Τυπώνει τα αποτελέσματα από τη «standard» έξοδο συνοδευόμενα με τα κατάλληλα μηνύματα.

ΔΟΜΗ

Προκειμένου να υλοποιηθεί το ζητούμενο χρησιμοποιήθηκαν, αρχικά, οι βιβλιοθήκες (.h) και οι εντολές :

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 τις έτοιμες συνάρτησεις «sin(...)», «cos(...)» και «fabs(...)» για τον υπολογισμό του ημιτόνου και το συνημιτόνου μιας γωνίας αντίστοιχα, σε ακτίνια και της απόλυτης τιμής ενός αριθμού.

c) «define»: Στη γραμμή «3» ορίζεται ο αναγνωριστικός αριθμός «pi» και η σταθερά «3.14159», όπου κατά την διάρκεια της προμεταγλώττισης του κώδικα θα αντικατασταθεί, όπου το «pi» η σταθερά «3.14159».

Επιπρόσθετα, χρησιμοποιήθηκαν οι χαρακτηριστικοί τελεστές :

a) αριθμητικοί : +, -, *, /

b) σχεσιακοί : <=, >

c) ανάθεσης : =

d) τελεστής & : Για την διεύθυνση μεταβλητής ως δεύτερο όρισμα της συνάρτησης «scanf()» που συνδέεται με τη «standard» είσοδο

e) μετααύξησης : μεταβλητή++

Οι εντολές ελέγχου :

a) if – else

Οι εντολές επαναλήψης :

a) do – while

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

b) for

ΜΕΤΑΒΛΗΤΕΣ

Ακέραιες μεταβλητές (τύπου «int»)

i (Η βοηθητική μεταβλητή για τον έλεγχο του δεύτερου βρόχου των γραμμών «30-33»)

j (Η βοηθητική μεταβλητή για τον υπολογισμό του δευτέρου όρου, του τρίτου, ...)

sign (Η βοηθητική μεταβλητή για την αλλαγή του προσήμου)

Πραγματικές μεταβλητές διπλής ακρίβειας (τύπου «double»)

deg (Η γωνία σε μοίρες)

rad (Η γωνία σε ακτίνια)

sine (Το ημίτονο της γωνίας σε ακτίνια)

term (Ο πρώτος όρος, ο δεύτερος ...)

next_term (Ο δεύτερος όρος, ο τρίτος ...)

diff_terms (Η διαφορά του δεύτερου όρου με τον πρώτο, του τρίτου με τον δεύτερο ...)

abs_diff_terms (Η απόλυτη τιμή της διαφοράς του δεύτερου όρου με τον πρώτο, του τρίτου με τον δεύτερο ...)

first_sin_T (Το άθροισμα του πρώτου με τον δεύτερο όρο με το κατάλληλο πρόσημο)

sin_T (Το άθροισμα του πρώτου με τον δεύτερο όρο, το άθροισμα αυτό με τον με τον τρίτο όρο ... με το κατάλληλο πρόσημο)

diff_sin_T (Η διαφορά του ημιτόνου με την απειροσειρά "Taylor")

abs_diff_sin_T (Η απόλυτη τιμή της διαφοράς του ημιτόνου με την απειροσειρά "Taylor")

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

ΔΙΑΣΧΙΣΗ

Δήλωση μεταβλητών

(γραμμές 9-16)

Το πρόγραμμα ξεκινάει με την δήλωση των μεταβλητών στις γραμμές «9-16» (αναλυτικά βλ. υποενότητα «Μεταβλητές»).

Αρχικοποίηση μεταβλητών

(γραμμές 13, 15, 16)

Στις προαναφερόμενες γραμμές γίνονται μερικές αρχικοποιήσεις μεταβλητών. Πιο αναλυτικά, στη γραμμή «13» η τιμή «0.0» εκχωρείται στη μεταβλητή «sin_T» για το άθροισμα του ημιτόνου της γωνίας σε ακτίνια με την απειροσειρά «Taylor», στη γραμμή «15» η τιμή «1» στη «j» ως βοηθητική μεταβλητή για τον υπολογισμό του δευτέρου όρου, του τρίτου κ.ο.κ. και στη «sign» η τιμή «-1» ως βοηθητική μεταβλητή για την αλλαγή του προσήμου στην πρόσθεση των όρων της απειροσειράς «Taylor».

Τίτλος προγράμματος

(γραμμή 19)

Στη γραμμή «19» με μία συναρτήση «printf()» τυπώνεται από τη «standard» έξοδο ο τίτλος του προγράμματος (Υπολογισμός του ημιτόνου μίας γωνίας), καθώς και δύο χαρακτήρες διαφυγής αυτής της αλλαγής γραμμής (\n).

Εισαγωγή της γωνίας σε μοίρες

(γραμμή 23)

Στη γραμμή «23» διαβάζεται από τη «standard» είσοδο με μία συνάρτηση «scanf()» μίας γωνία σε μοίρες συνοδευόμενο με τα κατάλληλα μηνύματα που τυπώνονται από τη «standard» έξοδο με τη συνάρτηση «printf()» (γραμμές 21, 22). Το αποτέλεσμα καταχωρείται στη μεταβλητή «deg».

Μετατροπή της γωνίας από μοίρες σε ακτίνια

(γραμμή 25)

Στη γραμμή «25» μετατρέπεται με την αριθμητική παράσταση «(pi * deg) / 180» η γωνία σε ακτίνια και το αποτέλεσμα καταχωρείται στη μεταβλητή «rad».

Εκτύπωση της γωνίας σε μοίρες

(γραμμή 26)

Στη γραμμή «26» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «deg» (η γωνία σε μοίρες) 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.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Εκτύπωση της γωνίας σε ακτίνια

(γραμμή 27)

Στη γραμμή «27» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «rad» (η γωνία σε ακτίνια) 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.

Υπολογισμός του ημιτόνου της γωνίας με τη συνάρτηση "sin (ω)"

(γραμμή 28)

Στη γραμμή «28» υπολογίζεται με την συνάρτηση «sin(rad)» που εισάγεται από την βιβλιοθήκη «math.h», το ημίτονο της γωνίας σε ακτίνια και το αποτέλεσμα καταχωρείται στη μεταβλητή «sine».

1ος Βρόχος

(γραμμές 29-52)

Στις γραμμές «29-52» υλοποιείται ένας βρόχος με την εντολή «do – while» με συνθήκη τερματισμού τη παράσταση «abs_diff_terms > 0.000001».

Πιο αναλυτικά, ο πρώτος βρόχος εκτελείται τουλάχιστον μία φορά και εφόσον, η παράσταση «abs_diff_terms > 0.000001» παράγει αποτέλεσμα μία τιμή «True» (συνθήκη αληθής) θα συνεχίσει να εκτελείται έως ότου παράγει αποτέλεσμα τη τιμή «False» (συνθήκη ψευδής). Η διάσχιση του «1^{ου} βρόχου» έχει ως εξής :

Αρχικοποίηση μεταβλητής του πρώτου όρου, του δευτέρου ...

(γραμμή 31)

Στη γραμμή «31» γίνεται μια αρχικοποίηση της μεταβλητής «term» που χαρακτηρίζει τον πρώτο όρο, τον δεύτερο κ.ο.κ της απειροσειράς, η οποία θα γίνεται κάθε φορά που εκτελείται ο «1^{ος} βρόχος». Πιο αναλυτικά, κάθε φορά που θα εκτελείται ο βρόχος στη «term» θα καταχωρείται η τιμή «1».

2^{ος} Βρόχος

(γραμμές 32-35)

Στις γραμμές «32-35» υλοποιείται ένας άλλος βρόχος μέσα στον πρώτο, με την εντολή «for» και με συνθήκη τερματισμού τη συνθήκη «i <= j», δηλαδή, για όσο η βοηθητική μεταβλητή ελέγχου του «2^{ου} βρόχου» είναι μικρότερη ή ίση της βοηθητικής μεταβλητής για τον υπολογισμό του δεύτερου όρου, του τρίτου κ.ο.κ. της απειροσειράς. Η βοηθητική «i» έχει για αρχική τιμή, την τιμή «1» και κάθε φορά που εκτελείται ο «2^{ος} βρόχος» η τιμή της αυξάνεται κατά ένα με την παράσταση «i++». Η διάσχιση του «2^{ου} βρόχου» έχει ως εξής :

Υπολογισμός του πρώτου όρου, του δευτέρου ...

(γραμμή 34)

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Στη γραμμή «34» υλοποιείται η παράσταση «term * (rad / i)» για τον υπολογισμό του πρώτου όρου, του δεύτερου κ.ο.κ. ($\omega^1 / 1!$, $\omega^3 / 3!$...) της απειροσειράς. Το αποτέλεσμα καταχωρείται στη μεταβλητή «term», κάθε φορά που εκτελείται ο «1^{ος} βρόχος» και ο «2^{ος} βρόχος». Η «term» σε κάθε εκτέλεση του πρώτου βρόχου έχει για αρχική τιμή, τη τιμή «1» (βλ. «Αρχειοποίηση μεταβλητής του πρώτου όρου, του δεύτερου ... (γραμμή 31)»).

Αύξηση της βοηθητικής μεταβλητής για τον υπολογισμό του δεύτερου όρου, του τρίτου ... (γραμμή 36)

Στη γραμμή «36» η βοηθητική μεταβλητή «j» για τον υπολογισμό του δεύτερου όρου, του τρίτου κ.ο.κ. της απειροσειράς, αυξάνεται κατά «2» με την παράσταση «j + 2», κάθε φορά που εκτελείται ο «1^{ος} βρόχος».

Υπολογισμός του δεύτερου όρου, του τρίτου ... (γραμμή 37)

Στη γραμμή «37» υλοποιείται η παράσταση «term * ((rad * rad) / (j * (j - 1)))» για τον υπολογισμό του δεύτερου όρου, του τρίτου κ.ο.κ. ($\omega^3 / 3!$, $\omega^5 / 5!$...) της απειροσειράς. Το αποτέλεσμα καταχωρείται στη μεταβλητή «next_term», κάθε φορά που εκτελείται ο «1^{ος} βρόχος».

Υπολογισμός της διαφοράς του δεύτερου όρου με τον πρώτο, του τρίτου με τον δεύτερο ... (γραμμή 38)

Στη γραμμή «38» καταχωρείται στη μεταβλητή «diff_terms» η διαφορά του δεύτερου όρου με τον πρώτο, του τρίτου με τον δεύτερο κ.ο.κ. της απειροσειράς (next_term – term), κάθε φορά που εκτελείται ο «1^{ος} βρόχος».

Υπολογισμός της απόλυτης τιμής της διαφοράς (γραμμή 39)

Στη γραμμή «39» καταχωρείται στη μεταβλητή «abs_diff_terms» η απόλυτη τιμή της διαφοράς του δεύτερου όρου με τον πρώτο, του τρίτου με τον δεύτερο κ.ο.κ. της απειροσειράς, κάθε φορά που εκτελείται ο «1^{ος} βρόχος». Η απόλυτη τιμή υπολογίζεται με τη βοήθεια της συνάρτησης «fabs(...)» που εισάγεται με τη βιβλιοθήκη «math.h».

(~) 1η επανάληψη του 1ου βρόχου (γραμμές 40-45)

(~) 2η, 3η ... επανάληψη του 1ου βρόχου (γραμμές 46-50)

Στις γραμμές «40-50» υλοποιείται ένας έλεγχος με την εντολή «if – else» για τον αριθμό επανάληψης του «1^{ου} βρόχου» που βρίσκεται το πρόγραμμα. Η συνθήκη ελέγχου είναι η παράσταση «j == 3» που ελέγχει το περιεχόμενο της βοηθητικής μεταβλητής «j» για τον υπολογισμό του δεύτερου όρου, του τρίτου κ.ο.κ. της απειροσειράς. Η συνθήκη ελέγχου πραγματοποιείται, διότι, οι χαρακτηριστικές πράξεις για τον υπολογισμό του ημιτόνου μιας γωνίας σε ακτίνια διαφέρουν μεταξύ τους από το δεύτερο άθροισμα και μετά. Προφανώς,

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

το άθροισμα του πρώτου με τον δεύτερο όρο είναι αυτό που διαφέρει ελάχιστα από τα άλλα αθροίσματα.

(~) 1η επανάληψη του 1ου βρόχου (γραμμές 40-45)

Στις γραμμές «89-94» εκτελούνται οι εντολές της «if (j == 3)» (γραμμή 40) για την περίπτωση που η «j» περιέχει την τιμή «3», δηλαδή, ότι το πρόγραμμα βρίσκεται στην πρώτη εκτέλεση του «1^{ου} βρόχου». Συνεπώς, υπολογίζεται το πρώτο άθροισμα της απειροσειράς που διαφέρει ελάχιστα από τα υπόλοιπα. Για «ω» γωνία σε ακτίνια υπολογίζεται η παράσταση $\omega^1 / 1! - \omega^3 / 3!$.

Υπολογισμός του πρώτου αθροίσματος με το κατάλληλο πρόσημο "sign" (γραμμή 42)

Στη γραμμή «42» υλοποιείται η παράσταση «term + (sign * next_term)» για το πρώτο άθροισμα (του πρώτου όρου με τον δεύτερο) με το κατάλληλο πρόσημο που καθορίζει η μεταβλητή «sign» που έχει για αρχική τιμή, τη τιμή «-1» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 16)»). Το αποτέλεσμα καταχωρείται στη μεταβλητή «first_sin_T».

Καταχώρηση του αθροίσματος στη μεταβλητή "sin T" (γραμμή 43)

Στη γραμμή «43» καταχωρείται με την παράσταση «sin_T + first_sin_T» το πρώτο άθροισμα με το κατάλληλο πρόσημο στη μεταβλητή «sin_T» που έχει για αρχική τιμή την τιμή «0.0» (βλ. «Αρχικοποίηση μεταβλητών (γραμμή 13)»).

Αλλαγή προσήμου (γραμμή 44)

Στη γραμμή «44» καταχωρείται στη μεταβλητή «sign» η χαρακτηριστική παράσταση «sign * (-1)» που βοηθάει στο να αλλάζει το πρόσημο της απειροσειράς αμέσως μετά το πρώτο άθροισμα και μετά.

(~) 2η, 3η ... επανάληψη του 1ου βρόχου (γραμμές 46-50)

Στις γραμμές «46-50» εκτελούνται οι εντολές της «else» (γραμμή 46) που αντιστοιχεί στην «if (j == 3)» (γραμμή 40), για την περίπτωση που η «j» δεν περιέχει την τιμή «3», δηλαδή, ότι το πρόγραμμα βρίσκεται στην δεύτερη εκτέλεση του «1^{ου} βρόχου» και μετά. Συνεπώς υπολογίζεται το δεύτερο άθροισμα, το τρίτο κ.ο.κ. της απειροσειράς. Για «ω» γωνία σε ακτίνια υπολογίζεται η παράσταση $(\omega^1 / 1! - \omega^3 / 3!) + \omega^5 / 5! \dots$.

Υπολογισμός του δευτέρου αθροίσματος, του τρίτου ... με το κατάλληλο πρόσημο "sign" (γραμμή 48)

Στη γραμμή «48» υλοποιείται η παράσταση «(sin_T + (sign * next_term))» για το δεύτερο άθροισμα, το τρίτο κ.ο.κ. της απειροσειράς με το κατάλληλο πρόσημο.

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

μο που καθορίζει η μεταβλητή «sign», όπου η «sign» έχει περιεχόμενο τη τιμή «1» μετά την πρώτη εκτέλεση του «1^{ου} βρόχου». Το αποτέλεσμα καταχωρείται στη μεταβλητή «sin_T», όπου η «sin_T» έχει περιεχόμενο το πρώτο άθροισμα μετά την πρώτη εκτέλεση του «1^{ου} βρόχου».

Αλλαγή προσήμου

(γραμμή 49)

Στη γραμμή «49» καταχωρείται στη μεταβλητή «sign» η χαρακτηριστική παράσταση «sign * (-1)» που βοηθάει στο να αλλάζει το πρόσημο της απειροσειράς αμέσως μετά το δεύτερο άθροισμα και μετά.

Εκτύπωση του ημιτόνου της γωνίας με την έτοιμη συνάρτηση "sin (ω)"

(γραμμή 54)

Στη γραμμή «54» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «sine» (το ημίτονο της γωνίας σε ακτίνια με την έτοιμη συνάρτηση «sin()») 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 αμέσως μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Εκτύπωση του ημιτόνου της γωνίας με την απειροσειρά "Taylor"

(γραμμή 55)

Στη γραμμή «55» τυπώνεται από τη «standard» έξοδο το περιεχόμενο της μεταβλητής «sin_T» (το ημίτονο της γωνίας σε ακτίνια με την απειροσειρά «Taylor») 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 αμέσως μετά από «20» κενούς χαρακτήρες με «6» δεκαδικά ψηφία, καθώς, πρόκειται για πραγματικό διπλής ακρίβειας. Αυτό αποσκοπεί στην ομοιόμορφη στοίχιση των αποτελεσμάτων.

Υπολογισμός της διαφοράς του ημιτόνου με την απειροσειρά "Taylor"

(γραμμή 56)

Στη γραμμή «56» υλοποιείται η παράσταση «sine – sin_T» για τον υπολογισμό της διαφοράς του ημιτόνου της γωνίας σε ακτίνιας με την έτοιμη συνάρτηση «sin()» (sine) και την απειροσειρά «Taylor» (sin_T). Το αποτέλεσμα καταχωρείται στη μεταβλητή «diff_sin_t»

Υπολογισμός της απόλυτης τιμής της διαφοράς του ημιτόνου με την απειροσειρά "Taylor"

(γραμμή 57)

Στη γραμμή «57» υλοποιείται η παράσταση «fabs (diff_sin_T)» για τον υπολογισμό της απόλυτης τιμής της διαφοράς του ημιτόνου της γωνίας σε ακτί-

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

για με την έτοιμη συνάρτηση «sin()» (sine) και την απειροσειρά «Taylor» (sin_T). Η συνάρτηση «fabs()» εισάγεται από τη βιβλιοθήκη «math.h». Το αποτέλεσμα καταχωρείται στη μεταβλητή «abs_diff_sin_t».

(~) Αποδεκτή απόλυτη τιμή της διαφοράς του ημιτόνου με την απειροσειρά "Taylor" (γραμμές 58-62)

(~) Απορριπτέα απόλυτη τιμή της διαφοράς του ημιτόνου με την απειροσειρά "Taylor" (γραμμές 63-67)

Στις γραμμές «58-67» υλοποιείται ένας έλεγχος με την εντολή «if – else» ως προς την απόκλιση της απόλυτης τιμής της διαφοράς των δύο συναρτήσεων που βγάζουν το ημίτονο της γωνίας. Η συνθήκη ελέγχου είναι η παράσταση «abs_diff_sin_T <= 0.0000009». Ο έλεγχος πραγματοποιείται για να διαπιστωθεί αν οι δύο αριθμοί είναι «σχεδόν» ίσοι.

(~) Αποδεκτή απόλυτη τιμή της διαφοράς του ημιτόνου με την απειροσειρά "Taylor" (γραμμές 58-62)

Στις γραμμές «58-62» εκτελούνται οι εντολές της «if (abs_diff_csts)» (γραμμή 58) για την περίπτωση που η απόλυτη τιμή της διαφοράς των δύο αριθμών είναι μικρότερη ή ίση από την τιμή «0.0000009» και επομένως, είναι «σχεδόν» ίσοι. Εάν, η παράσταση «abs_diff_sin_T <= 0.0000009» παράγει αποτέλεσμα μία τιμή «True» (τιμή διάφορη του «0»), τότε, θα τυπωθούν από τη «standard» έξοδο με τη συνάρτηση «printf()» τα χαρακτηριστικά μηνύματα.

(~) Απορριπτέα απόλυτη τιμή της διαφοράς του ημιτόνου με την απειροσειρά "Taylor" (γραμμές 63-67)

Στις γραμμές «63-67» εκτελούνται οι εντολές της «else» (γραμμή 63) που αντιστοιχεί στην «if (abs_diff_sin_T)» (γραμμή 58) για την περίπτωση που η απόλυτη τιμή της διαφοράς των δύο αριθμών είναι μεγαλύτερη από την τιμή «0.0000009» και επομένως, οι δύο αριθμοί δεν είναι «σχεδόν» ίσοι. Εάν, η παράσταση «abs_diff_sin_T <= 0.0000009» παράγει αποτέλεσμα τη τιμή «False» (0), τότε, θα τυπωθούν από τη «standard» έξοδο με τη συνάρτηση «printf()» τα χαρακτηριστικά μηνύματα.

ΠΑΡΑΔΕΙΓΜΑΤΑ

Παράδειγμα 1 ($\omega == 30^\circ$)

=====

Υπολογισμός του ημιτόνου μίας γωνίας

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

=====

Εισάγετε γωνία στο διάστημα του 1ου κύκλου [0,360]

Μοίρες : 30

Μοίρες : [30.000000]

Ακτίνια : [0.523598]

Sine : [0.500000]

Taylor : [0.500000]

Sine ~= Taylor

Οι δύο αριθμοί είναι σχεδόν ίσοι

Παράδειγμα 2 ($\omega == 360^\circ$)

=====

Υπολογισμός του ημιτόνου μίας γωνίας

=====

Εισάγετε γωνία στο διάστημα του 1ου κύκλου [0,360]

Μοίρες : 360

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Μοίρες : [360.000000]

Ακτίνια : [6.283180]

Sine : [-0.000005]

Taylor : [-0.000005]

Sine ~= Taylor

Οι δύο αριθμοί είναι σχεδόν ίσοι

Παράδειγμα 3 ($\omega == 167.5^\circ$)

=====

Υπολογισμός του ημιτόνου μίας γωνίας

=====

Εισάγετε γωνία στο διάστημα του 1ου κύκλου [0,360]

Μοίρες : 167.5

Μοίρες : [167.500000]

Ακτίνια : [2.923424]

Sine : [0.216442]

Taylor : [0.216442]

Sine ~= Taylor

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

Οι δύο αριθμοί είναι σχεδόν ίσοι

ΠΑΡΑΤΗΡΗΣΕΙΣ

Η ορθότητα των αποτελεσμάτων στα παραπάνω παραδείγματα ελέγχθηκε και με την χρήση τυπικής αριθμομηχανής και διαπίστωθηκε μία διαφορά στο «Παράδειγμα 2 ($\omega == 360^\circ$)».

Αναλυτικά, έχουμε «3» παραδείγματα που αντιστοιχούν σε γωνία που ανήκει στο διάστημα του πρώτου κύκλου $[0, 360]$.

Παράδειγμα 1 ($\omega == 30^\circ$)

Στο «Παράδειγμα 1» διαβάζεται από τη «standard» είσοδο η γωνία των «30» μοιρών και ορθά μετατρέπεται σε ακτίνια, υπολογίζεται το ημίτονο της με την έτοιμη συνάρτηση « $\sin(\dots)$ » και την απειροσειρά «Taylor». Οι δύο αριθμοί είναι «σχεδόν» ίσοι ως προς το «7^ο» δεκαδικό τους ψηφίο, όπου άμα πάρουμε την απόλυτη τιμή της διαφοράς τους θα διαπιστωθεί ότι είναι μικρότερη από την τιμή «0.0000009». Τα αποτελέσματα :

Μοίρες : 30.000000

Ακτίνια : 0.523598

$\sin(\dots)$: 0.500000

Taylor : 0.500000

Παράδειγμα 2 ($\omega == 360^\circ$)

Στο «Παράδειγμα 2» διαβάζεται από τη «standard» είσοδο η γωνία των «360» μοιρών και ορθά μετατρέπεται σε ακτίνια, υπολογίζεται το ημίτονο της με την έτοιμη συνάρτηση « $\sin(\dots)$ » και την απειροσειρά «Taylor». Οι δύο αριθμοί είναι «σχεδόν» ίσοι ως προς το «7^ο» δεκαδικό τους ψηφίο, όπου άμα πάρουμε την απόλυτη τιμή της διαφοράς τους θα διαπιστωθεί ότι είναι μικρότερη από την τιμή «0.0000009». Τα αποτελέσματα :

Μοίρες : 360.000000

Ακτίνια : 6.283180

$\sin(\dots)$: -0.000005

Taylor : -0.000005

Η παρατήρηση είναι ότι στο ημίτονο το αποτέλεσμα που βγάζουν και οι δύο

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

αριθμοί, ότι είναι η τιμή «-0.000005», ενώ η τυπική αριθμομηχανή βγάζει ότι το ημίτονο των «360» μοιρών είναι η τιμή «0.000000».

Παράδειγμα 3 ($\omega == 167.5^\circ$)

Στο «Παράδειγμα 3» διαβάζεται από τη «standard» είσοδο η γωνία των «167.5» μοιρών και ορθά μετατρέπεται σε ακτίνια, υπολογίζεται το ημίτονο της με την έτοιμη συνάρτηση «sin(...)» και την απειροσειρά «Taylor». Οι δύο αριθμοί είναι «σχεδόν» ίσοι ως προς το «7^ο» δεκαδικό τους ψηφίο, όπου άμα πάρουμε την απόλυτη τιμή της διαφοράς τους θα διαπιστωθεί ότι είναι μικρότερη από την τιμή «0.0000009». Τα αποτελέσματα :

Μοίρες : 167.500000

Ακτίνια : 2.923424

sin(...) : 0.216442

Taylor : 0.216442

ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ



Σας ευχαριστώ για την προσοχή σας.

