

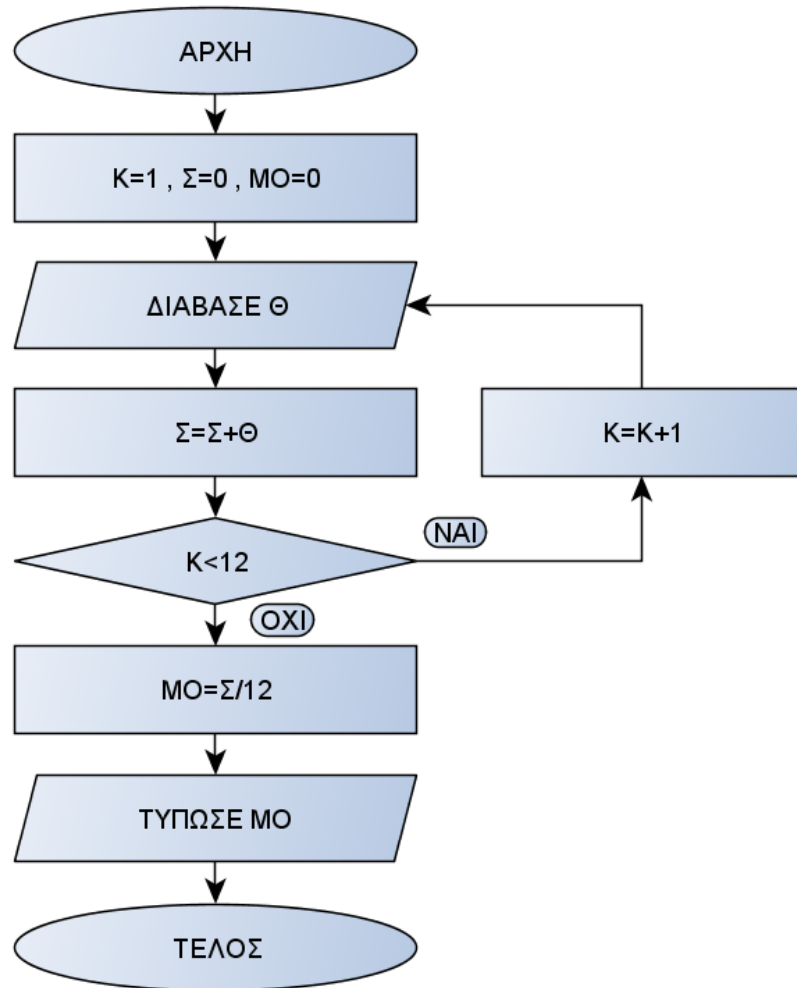
# Εισαγωγή στη γλώσσα προγραμματισμού C++

Επαναληπτική Δομή  
2023

Άριστος Πασιάς

# 1. Εισαγωγή

2



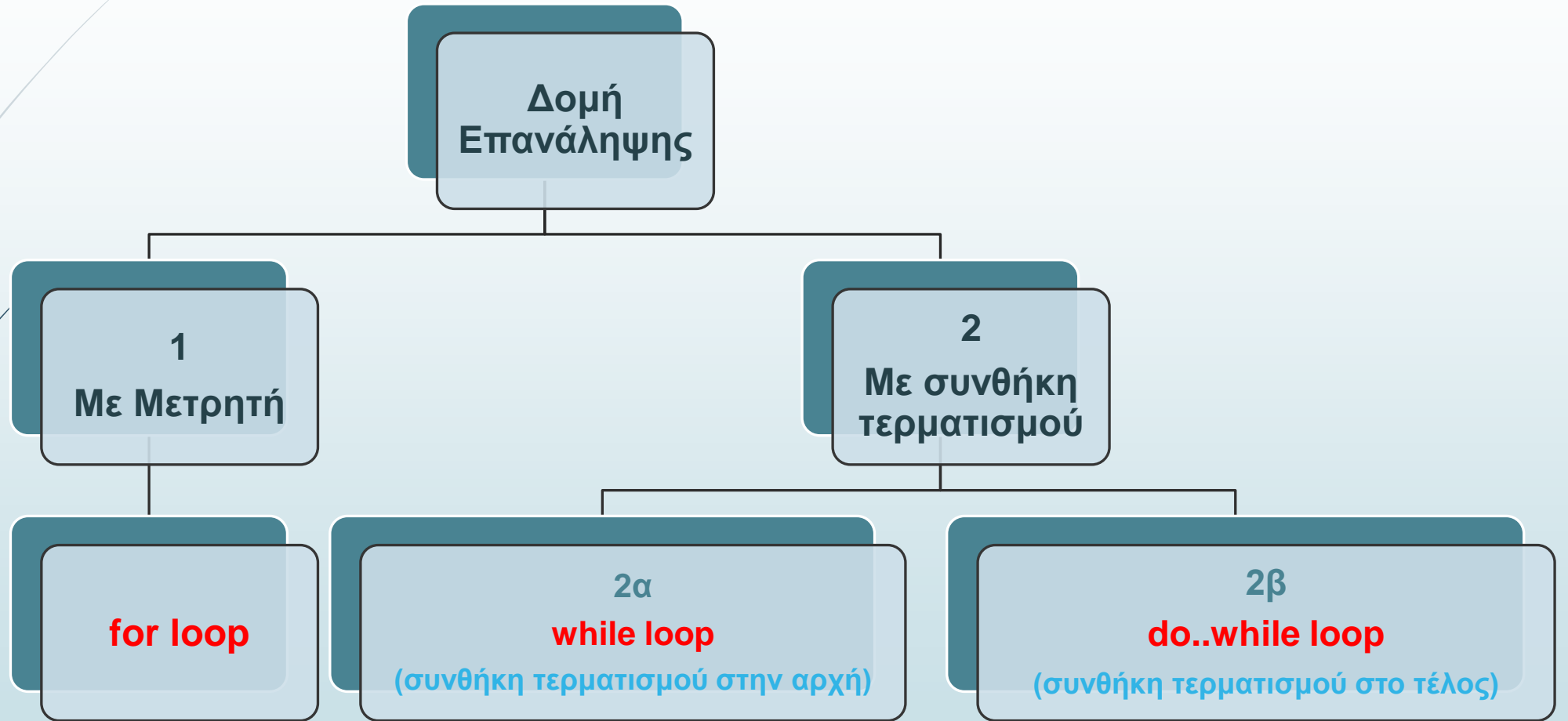
Δομές επανάληψης ή βρόχοι (Loops) ονομάζονται τμήματα του κώδικα που εκτελούνται περισσότερες από μία φορές, ανάλογα με τη συνθήκη που έχουμε δηλώσει.

Οι βρόχοι αναγκάζουν ένα τμήμα κώδικα να επαναλαμβάνεται. Η επανάληψη συνεχίζεται για όσο μία λογική συνθήκη είναι αληθής. Όταν η συνθήκη είναι ψευδής, ο βρόχος τελειώνει και ο έλεγχος του προγράμματος συνεχίζεται παρακάτω.

# Οι πιο κοινές χρήσεις βρόχων (Loops)

- Για καταμέτρηση
- Για τη συσσώρευση
- Για αναζήτηση
- Για ταξινόμηση
- Για την προβολή πινάκων
- Για την καταχώρηση δεδομένων
- Για εμφάνιση μενού
- Για επεξεργασία λίστας
- Για προσομοιώσεις

# Εντολές Επανάληψης της C++:



# Ας θυμηθούμε...

► Τελεστές αύξησης/μείωσης ( $++$  ,  $--$ )

- $a++$ ;
- $a += 1$ ;
- $a = a + 1$ ;

❖  $b = 4$ ;

❖  $a = ++b$ ; //  $a = 5$  και  $b = 5$

❖  $a = b++$ ; //  $a = 4$  και  $b = 5$

**$++b$**  Πρώτα αυξάνεται η τιμή της μεταβλητής και μετά γίνονται οι πράξεις

**$b++$**  Πρώτα γίνονται οι πράξεις και μετά αυξάνεται η τιμή της μεταβλητής.

# Βρόχος **for**

1. **for** (αρχική τιμή ; Συνθήκη επανάληψης ; Βήμα)
2. {
3. Εντολές επανάληψης
4. }

Τα άγκιστρα μπορούν να παραλείπονται, αν έχουμε μόνο μία εντολή.

- Ο βρόχος **for** χρησιμοποιείται όταν γνωρίζουμε πριν από την έναρξη του βρόχου τον ακριβή αριθμό των επαναλήψεων.
- Ξεκινά με τη δεσμευμένη λέξη **for** και ακολουθούν μέσα σε παρένθεση οι οδηγίες εκτέλεσης του βρόχου. Αποτελούνται από τρία τμήματα που χωρίζονται μεταξύ τους με το σύμβολο `;`
  1. **Αρχική τιμή:** Ορίζεται μία μεταβλητή που παίζει τον ρόλο του μετρητή των επαναλήψεων. Εδώ, η μεταβλητή-μετρητής λαμβάνει αρχική τιμή.
  2. **Συνθήκη επανάληψης:** Ακολουθεί μία λογική συνθήκη. Όσο η λογική συνθήκη είναι αληθής, ο βρόχος θα επαναλαμβάνει τις εντολές που περιέχονται ανάμεσα στα άγκιστρα `{ ... }`. Όταν γίνει ψευδής, ο έλεγχος του προγράμματος συνεχίζει με τη πρώτη εντολή μετά τον βρόχο. Η εξέταση της λογικής συνθήκης γίνεται πριν ο έλεγχος του προγράμματος εισέλθει στον βρόχο.
  3. **Βήμα:** Η τελευταία παράμετρος μέσα στις παρενθέσεις είναι το βήμα μεταβολής του μετρητή.

# Παράδειγμα 1 for pxed01

7

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να εμφανίζει όλους τους αριθμούς από το 1 μέχρι το 10.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i;
8      for (i = 1; i <= 10; i++)
9      {
10         cout << i << endl;
11     }
12     return 0;
13 }
14
```

Αρχική τιμή μετρητή

Συνθήκη  
Επανάληψης

Βήμα  
αύξησης/μείωσης του  
μετρητή

**Ερώτηση:** πως θα μπορούσα να τυπώσω τους αριθμούς σε φθίνουσα σειρά;

```
1
2
3
4
5
6
7
8
9
10
```

Process re  
Press any

# Παράδειγμα 2 for pxed02

8

- Να δημιουργήσετε ένα πρόγραμμα σε C++ που να εμφανίζει όλα τα ακέραια πολλαπλάσια του 5 που είναι μεγαλύτερα του 100 και μικρότερα του 1000.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i;
8      for (i = 105; i < 1000; i+=5)
9      {
10         cout << i << endl;
11     }
12     return 0;
13 }
14
```

```
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
```

# Παράδειγμα 3 for pxed03

9

Να δημιουργήσετε ένα πρόγραμμα σε C++ που να εμφανίζει όλα τα ακέραια πολλαπλάσια του 5 και το προοδευτικό τους άθροισμα που είναι μεγαλύτερα του 100 και μικρότερα του 1000.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i,s=0;
8      for (i = 105; i < 1000; i+=5)
9      {
10         s=s+i;
11         cout << i << " " << s << endl;
12     }
13     return 0;
14 }
15
```

```
105  105
110  215
115  330
120  450
125  575
130  705
135  840
140  980
145  1125
150  1275
```

# Παράδειγμα 4 for pxed04

10

Να δημιουργήσετε πρόγραμμα το οποίο να διαβάζει τη βάση ( $b \geq 0$ ) και τον εκθέτη ( $e \geq 0$ ) μίας δύναμης και να επιστρέφει ως αποτέλεσμα τη βάση υψωμένη στον εκθέτη. Π.χ.  $2^3 = 8$ . (Χωρίς τη χρήση έτοιμης μαθηματικής συνάρτησης)

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int b,e,i,result = 1;
8      cout << "Insert the base number ";
9      cin >> b;
10     cout << "insert the exponent";
11     cin >> e;
12     for (i = 1; i <= e; i++) {
13         result = result * b;
14     }
15     cout << result << endl;
16     return 0;
17 }
18
```

```
Insert the base number 2
insert the exponent7
128
```

# Παράδειγμα 5 for

- Με τη βοήθεια προκαταρκτικής εκτέλεσης, να γράψετε το αναμενόμενο αποτέλεσμα του πιο κάτω προγράμματος:

```
#include <iostream>
using namespace std;
int main(){
int mycount;
for (mycount=1;mycount <= 3; mycount++)
cout << "Hello" << endl;
cout << "Good Day" << endl;
cout << "so long" << endl;
return 0;
}
```

Λύση:

hello

hello

hello

good day

so long

# Παράδειγμα 6 for

- Με τη βοήθεια προκαταρκτικής εκτέλεσης, να γράψετε το αναμενόμενο αποτέλεσμα του πιο κάτω προγράμματος:

```
#include <iostream>
using namespace std;
int main(){
int mycount;
for (mycount=1;mycount <= 3; mycount++) {
cout << "Hello" << endl;
cout << "Good Day" << endl;
cout << "so long" << endl;
}
return 0;
}
```

## Λύση:

```
Hello
good day
so long
Hello
good day
so long
hello
good day
so long
```

# Παράδειγμα 7 for

13

Με τη βοήθεια προκαταρτικής εκτέλεσης, να γράψετε το αναμενόμενο αποτέλεσμα του πιο κάτω προγράμματος:

```
#include <iostream>
using namespace std;
int main()
{
    int mycount, mysum;

    mysum = 0;
    for (mycount = 1 ;mycount <= 4; mycount++)
    {
        mysum = mysum + mycount;
    }
    cout << ("ΤΟ ΑΘΡΙΣΜΑ ΕΙΝΑΙ: ") << mysum <<
endl;
    system("Pause");
    return 0;
}
```

Άριστος Πασιάς

Προκαταρτική Εκτέλεση	
mycount	mysum
1	0+1=1
2	1+2=3
3	3+3=6
4	6+4=10

Αποτέλεσμα : sum= 10

# Άσκηση 1 for exedf01

14

Να δημιουργήσετε πρόγραμμα το οποίο να διαβάζει δέκα τυχαίους θετικούς αριθμούς, μικρότερους του 1000 και να εμφανίζει: (α) τον μεγαλύτερο, (β) τον μικρότερο, (γ) το άθροισμα και (δ) τον Μέσο όρο.

```
Insert the 1number
3
Insert the 2number
4
Insert the 3number
3
Insert the 4number
4
Insert the 5number
3
Insert the 6number
4
Insert the 7number
3
Insert the 8number
4
Insert the 9number
3
Insert the 10number
4
Minimum= 3 maximum= 4 Summary= 35 MO= 3.5
Process returned 0 (0x0)   execution time : 7.485 s
Press any key to continue.
```

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int t, a=0, elax=1000, meg=0, athr=0;
8      float mo=0.0;
9      for(int i=1;i<=10;i++) {
10         a=0;
11         cout<< "Insert the "<<i<<"number"<<endl;
12         cin >> a;
13         athr=athr+a;
14         if (a>meg) {
15             meg=a;
16         }
17         if (a<elax) {
18             elax=a;
19         }
20         t=i;
21     }
22     mo=(float) athr/t;
23     cout << "Minimum= " <<elax<<" maximum= " <<meg<<
24         |" Summary= " <<athr<<" MO= " <<mo<<endl;
25     return 0;
26 }
27
```

## Άσκηση 2 for exedf02

15

Μια εταιρεία εργοδοτεί διάφορους ωρομισθίους υπαλλήλους. Στο τέλος κάθε ημέρας ο διευθυντής της εταιρείας θέλει να εισάγει σε ένα πρόγραμμα τον αριθμό των υπαλλήλων που εργάστηκαν εκείνη τη μέρα ( $n$ ), το ποσό της αμοιβής ανά ώρα εργασίας εκείνης της μέρας ( $k$ ) και τον αριθμό των ωρών που εργάστηκε κάθε υπάλληλος ( $h$ ). Το πρόγραμμα πρέπει να υπολογίζει τον ακάθαρτο μισθό κάθε υπαλλήλου ( $am=k*h$ ), τον φόρο που αναλογεί σε κάθε υπάλληλο ( $f=am*r/100$  : όπου  $r$  το ποσοστό φορολογίας που είναι  $0$  για τα πρώτα  $100€$  μισθού,  $2\%$  για τα επόμενα  $100€$  μισθού και  $5\%$  για όλο το υπόλοιπο ποσό μισθού.), το καθαρό μισθό κάθε υπάλληλου ( $km=am-f$ ), το συνολικό ποσό που πρέπει να πληρώσει στους υπαλλήλους ( $tkm=sum(km)$ ) και στο φόρο ( $tf=2(sum(f))$ ).

Να δημιουργήσετε το κατάλληλο πρόγραμμα. Στο τέλος να εμφανίζει το όνομα κάθε υπαλλήλου, τον ακάθαρτο μισθό του, τον φόρο που πρέπει να του αποκοπεί το καθαρό ποσό που θα πάρει μετά την αποκοπή του φόρου, το σύνολο των μισθών και των φόρων.

# Άσκηση 2 for δείγμα αποτελέσματος

16

```
Insert the number of employs
3
Insert salary per hour
20
insert name and working hours of the 1 employ
alfa 5
onoma: Gross salary: foros: Net salary:
alfa          100      0      100

insert name and working hours of the 2 employ
Vita 8
onoma: Gross salary: foros: Net salary:
Vita          160     1.2     158.8

insert name and working hours of the 3 employ
Gamma 15
onoma: Gross salary: foros: Net salary:
Gamma         300      7      293

Total Gross salaries: 560 Total Foros: 16.4 Total Net: 551.8

Process returned 0 (0x0)   execution time : 63.588 s
Press any key to continue.
```

# Άσκηση 2 for υπόδειγμα λύσης

17

```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main()
6  {
7      int n;
8      string name;
9      float h=0, k=0, am=0, f=0, km=0, tam=0, tkm=0, tf=0;
10     cout<<"Insert the number of employs "<<endl;
11     cin>>n;
12     cout<<"Insert salary per hour "<<endl;
13     cin>>k;
14     for(int i=1;i<=n;i++){
15         cout<<"insert name and working hours of the "<< i<<" employ "<<endl;
16         cin>>name>>h;
17         am=h*k;
18         if (am<=100){
19             f=0;
20         }
21         else if(am<=200){
22             f=(am-100)*0.02;
23         }
24         else {
25             f=2+(am-200)*0.05;
26         }
27         km=am-f;
28         tam=tam+am;
29         tf=tf+2*f;
30         tkm=tkm+km;
31         cout<<"onoma: "<<" Gross salary: "<<" foros: "<<" Net salary: "<<endl;
32         cout<<name<<"          "<<am<<"          "<<f<<"          "<<km<<endl<<endl;
33     }
34     cout<<endl<<endl;
35     cout << "Total Gross salaries: " <<tam<<" Total Foros: "<<tf<<" Total Net: "<<tkm<<endl<<endl;
36     return 0;
37 }
```

# Άσκηση 3 for exedf03

18

Να δημιουργήσετε πρόγραμμα σε C++ το οποίο να δέχεται ένα ακέραιο θετικό αριθμό μικρότερο του 100 και να τυπώνει όλους τους ακέραιους αριθμούς που είναι μικρότεροι από N και μεγαλύτεροι από N/2.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i, n;
8      cout << "Insert an integer below 100 " << endl;
9      cin >> n;
10     int k=n/2;
11     for (i=k+1; i<n; i++) {
12         cout <<" |"<<i;}
13     return 0;
14 }
```

```
Insert an integer below 100
12
7 8 9 10 11
```

```
Insert an integer below 100
15
8 9 10 11 12 13 14
```

# Άσκηση 4 for exedf04

19

Να δημιουργήσετε πρόγραμμα σε C++ το οποίο να δέχεται ένα ακέραιο θετικό αριθμό και να τυπώνει τον πίνακα πολλαπλασιασμού του αριθμού μέχρι κάποιο δοσμένο όριο.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n, r;
6      cout << "Enter a positive integer: ";
7      cin >> n;
8      cout << "Enter range: ";
9      cin >> r;
10     for (int i = 1; i <= r; ++i) {
11         cout << n << " * " << i << " = " << n * i << endl;
12     }
13
14     return 0;
15 }
```

```
Enter a positive integer: 8
Enter range: 12
8 * 1 = 8
8 * 2 = 16
8 * 3 = 24
8 * 4 = 32
8 * 5 = 40
8 * 6 = 48
8 * 7 = 56
8 * 8 = 64
8 * 9 = 72
8 * 10 = 80
8 * 11 = 88
8 * 12 = 96
```

# Άσκηση 5 for exedf05

20

- Ο πληθυσμός μιας αποικίας μυρμηγκιών αυξάνεται κατά 15% κάθε μήνα. Να δημιουργήσετε ένα πρόγραμμα το οποίο να διαβάσει τον αρχικό πληθυσμό της αποικίας και να υπολογίζει και να εμφανίζει την μηνιαία αύξηση και τον πληθυσμό της αποικίας για ένα αριθμό μηνών που θα δίνεται από τον χρήστη.

```
1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int i,p,m,a=0;
6      const float inp=0.15;
7      cout<<"Αρχικος Plithismos "<<endl;
8      cin>>p;
9      cout<<"Gia posoys mines? "<<endl;
10     cin>>m;
11     cout<<"\t minas"<<" \t Afxisi "<<" \t plithismos "<<endl;
12     for (i=0;i<=m;i++) {
13         cout<<"\t "<<i<<" \t "<<a<<" \t \t "<<p<<endl;
14         a=p*inp;
15         p=p+a;
16     }
17     cout << "T E L O S !" << endl;
18     return 0;
19 }
```

```
kos Plithismos
00
posoys mines?

minas      Afxisi      plithismos
0           0           800000
1           7999          807999
2           8079          816078
3           8160          824238
4           8242          832480
5           8324          840804
6           8408          849212
7           8492          857704
8           8577          866281
9           8662          874943
10          8749          883692
11          8836          892528
12          8925          901453

O S !
```

# Άσκηση exedw01

21

- Να δημιουργήσετε ένα πρόγραμμα στο οποίο να εισάγουμε διαδοχικά από το πληκτρολόγιο διάφορους αριθμούς από το 1 μέχρι το 100 και να τους αθροίζει. Όταν το άθροισμα γίνει ίσο ή μεγαλύτερο από 100 θα σταματά και θα τυπώνει το άθροισμα που υπολόγισε.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      float a,s;
8      while(s<=100) {
9          cout<<" Insert a number between 1 and 100 "<<endl;
10         cin >>a;
11         s=s+a;
12     }
13     cout << "The sum = " <<s<< endl;
14     return 0;
15 }
16
```

Insert a number between 1 and 100

ime : 10.755 s

while

# Βρόχος While

22

- Χρησιμοποιούμε τον βρόχο **while** όταν θέλουμε να επαναλαμβάνεται ένα μπλοκ εντολών ανάλογα με την τιμή μιας λογικής συνθήκης η οποία αλλάζει τιμή κατά την εκτέλεση των εντολών επανάληψης. Συνήθως δεν γνωρίζουμε από πριν τον αριθμό των επαναλήψεων.
- Ο βρόχος **while**, ελέγχει τη λογική συνθήκη πριν την έναρξη του βρόχου και εάν αυτή είναι αληθής, εκτελεί τις εντολές που βρίσκονται ανάμεσα στα άγκιστρα **{ ... }**. Αυτή η διαδικασία επαναλαμβάνεται όσο η λογική συνθήκη παραμένει αληθής.
- Όταν η λογική συνθήκη γίνει ψευδής, τότε ο έλεγχος του προγράμματος θα συνεχίσει με το υπόλοιπο πρόγραμμα μετά από τον βρόχο.

1. **while** (**Συνθήκη**)
2. {
3.     **Εντολές επανάληψης**
4. }
5.     **Εντολή μετά τον βρόχο**

## Παράδειγμα:

```
int counter = 1;
while (counter <= 10) {
    cout<<"count = "<<counter<<endl;
    counter++;
}
next statement;
```

# Βρόχος While

23

- Η συνθήκη, που βρίσκεται στην αρχή του βρόχου (Loop), ελέγχεται πριν εκτελεστεί η εντολή ή το σώμα των εντολών και αν το αποτέλεσμα είναι false, τότε δεν εκτελούνται καθόλου.
- Επομένως, για να λειτουργήσει ο βρόχος θα πρέπει η συνθήκη να έχει αρχικά τιμή true.
- Σε κάθε επανάληψη ελέγχεται η συνθήκη και αν είναι true ο βρόχος εκτελείται, διαφορετικά τελειώνει και εκτελείται η επόμενη εντολή. Επομένως η τιμή της συνθήκης θα πρέπει να μεταβάλλεται μέσα στον βρόχο.
- Χρειάζεται ιδιαίτερη προσοχή στον καθορισμό της συνθήκης ώστε να αποφεύγεται η δημιουργία ατέρμονα βρόχου (**infinite loop**).

# Παράδειγμα 1 While pxed01

24

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να διαβάζει από μια ζυγαριά το βάρος από διάφορα κιβώτια και θα το αθροίζει μέχρι να συμπληρώσει συνολικό βάρος μεγαλύτερο ή ίσο με 100.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, s=0;
8      cout << "Insert weight " << endl;
9      while (s < 100){
10         cin >> n;
11         s=s+n;
12         cout << "Weight until now: " << s << endl;
13     }
14     cout << "Total weight is " << s;
15     return 0;
16 }
17
```

```
Insert weight
20
Weight until now: 20
22
Weight until now: 42
12
Weight until now: 54
21
Weight until now: 75
18
Weight until now: 93
12
Weight until now: 105
Total weight is 105
```

## Παράδειγμα 2 While exedw03

25

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να διαβάζει από μια ζυγαριά το βάρος από διάφορα κιβώτια και θα το αθροίζει μέχρι να συμπληρώσει συνολικό βάρος μεγαλύτερο ή ίσο με 100. Στο τέλος να εμφανίζει το συνολικό βάρος και τον αριθμό των κιβωτίων που χρειάστηκε να αθροίσει.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, i=0, s=0;
8      cout << "Insert weight " << endl;
9      while (s < 100){
10         cin >> n;
11         s=s+n;
12         i++;
13         cout << "Weight until now: " << s << endl;
14     }
15     cout << "Total weight is " << s << " In "
16     <<i<<" boxes"<<endl;
17     return 0;
18 }
19
```

```
Insert weight
24
Weight until now: 24
12
Weight until now: 36
25
Weight until now: 61
16
Weight until now: 77
20
Weight until now: 97
12
Weight until now: 109
Total weight is 109 In 6 boxes
```

# Παράδειγμα 3 While exedw04

26

Να δημιουργήσετε ένα πρόγραμμα το οποίο να υπολογίζει το τετράγωνο και την τετραγωνική ρίζα όλων των ζυγών θετικών αριθμών που είναι μικρότεροι από 10.

```
1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4  #include <stdio.h>
5  #include <stdlib.h>
6  using namespace std;
7
8  int main()
9  {
10     float n, sq, sqr;
11     n = 2;
12     while (n <= 10)
13     {
14         sq = pow(n, 2);
15         sqr = sqrt(n);
16         cout<<"Number: " <<fixed<<setprecision(2)<<
17         |n<<"\t Tetragono " <<sq<<"\t Riza: " <<sqr<<endl;
18         n = n + 2;
19     }
20     cout << "The End"<< endl ;
21     system("Pause");
22     return (0);
23 }
24
```

```
Number: 2.00    Tetragono 4.00    Riza: 1.41
Number: 4.00    Tetragono 16.00    Riza: 2.00
Number: 6.00    Tetragono 36.00    Riza: 2.45
Number: 8.00    Tetragono 64.00    Riza: 2.83
Number: 10.00   Tetragono 100.00    Riza: 3.16
The End
Press any key to continue . . .
```


# While instead of for

27

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να εμφανίζει όλους τους αριθμούς από το 1 μέχρι το 10. (α) με την εντολή While. (β) Με την εντολή for.

## While


```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i=1;
8     while (i<=10)
9     {
10    cout << i << endl;
11    i++;
12    }
13    return 0;
14 }
15
```



## Vs

## for

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main()
6 {
7     int i;
8     for (i = 1; i <= 10; i++)
9     {
10    cout << i << endl;
11    }
12    return 0;
13 }
14
```



# Παράδειγμα 5 While exedw05

28

- Ο πληθυσμός μιας αποικίας μυρμηγκιών αυξάνεται κατά 15% κάθε μήνα. Ταυτόχρονα 10000 μυρμηγκία πεθαίνουν το ίδιο χρονικό διάστημα. Να δημιουργήσετε ένα πρόγραμμα το οποίο να διαβάζει τον αρχικό πληθυσμό της αποικίας και να υπολογίζει και να εμφανίζει τον αριθμό των μηνών που θα χρειαστεί η αποικία να ξεπεράσει το 1 000 000 πληθυσμό.

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int pop, month=0;
8      cout << "Insert the population on zero month" << endl;
9      cin >> pop;
10     while(pop<=1000000) {
11         month++;
12         pop=pop*1.15-10000;
13     }
14     cout << "They need " << month << " months"<< endl;
15     return 0;
16 }
17
```

```
Insert the population on zero month
90000
They need 27 months
```

# Άσκηση For ή While exedfw06

29

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα σε C++ που να δέχεται ένα ακέραιο αριθμό μικρότερο από 100 και να υπολογίζει το άθροισμα όλων των ακέραιων αριθμών από το 1 μέχρι τον αριθμό αυτό. Πχ αν δοθεί ο αριθμός 5 να υπολογίζει το άθροισμα  $1+2+3+4+5 = 15$

- A. Να δώσετε μια λύση με την χρήση της εντολής βρόχου for
- B. Να δώσετε μια λύση με την χρήση της εντολής βρόχου While
- Γ. Να συμπληρώσετε το πρόγραμμα με την δυνατότητα ελέγχου των δεδομένων εισόδου

```
1  #include <iostream>
2
3  using namespace std;
4  // Find 1+2+3+...n
5  int main()
6  {
7      int n, counter = 1, sum = 0;
8      cout << "\n Enter limit: " << endl;
9      cin>>n;
10     while(n<1||n>100){
11         cout << "\n You have to insert a number from 1 to 100. Try again " << endl;
12         cin>>n;
13     }
14     while (counter <= n){
15         sum = sum + counter;
16         counter = counter + 1;
17     }
18     cout<<"\nThe sum is " <<sum;
19     return 0;
20 }
21
```

# Βρόχος Do

30

- Σε μερικές περιπτώσεις θέλουμε να εξασφαλίσουμε ότι οι εντολές κάποιου βρόχου θα εκτελεστούν μία τουλάχιστον φορά, ανεξάρτητα από την τιμή της λογικής συνθήκης.
- Σε αυτή την περίπτωση χρησιμοποιούμε τον βρόχο **do**, ο οποίος κάνει τον έλεγχο της λογικής συνθήκης στο **τέλος του βρόχου** σε αντίθεση με τον βρόχο while ο οποίος κάνει τον έλεγχο πριν ξεκινήσει να εκτελεί τις εντολές.
- **Προσοχή** στο ερωτηματικό που υπάρχει μετά τη συνθήκη, το οποίο σηματοδοτεί το τέλος της πρότασης.
- Όσο η λογική συνθήκη είναι αληθής, ο βρόχος θα εκτελεί τις εντολές του. Όταν η λογική συνθήκη γίνει ψευδής, ο έλεγχος του προγράμματος θα συνεχίσει παρακάτω.

## Βρόχος do While

1. **do {**  
    **Εντολές επανάληψης**
2. **}**
3. **while (Συνθήκη) ;**
4. **Εντολή μετά τον βρόχο**

# Παράδειγμα 1 do pxedd01

31

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να δέχεται έναν ακέραιο αριθμό και να τον παρουσιάζει στην οθόνη. Το πρόγραμμα θα σταματήσει να δέχεται αριθμούς, όταν δοθεί ο αριθμός 0.

```
1      #include <iostream>
2
3      using namespace std;
4
5      int main()
6      {
7          int n;
8          do
9          {
10         cin >>n;
11         cout << "You pressed: " << n << endl;
12     }
13     while (n!=0);
14     return 0;
15 }
16
```

# Παράδειγμα do while pxeddw01

32

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να δέχεται ακέραιους αριθμούς και να παρουσιάζει στην οθόνη το τετράγωνο και τον κύβο τους. Το πρόγραμμα θα σταματήσει να δέχεται αριθμούς, όταν δοθεί ο αριθμός 0.

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      int n, n2=0,n3=0;
8      cout<<"insert a number "<<endl;
9      cin>>n;
10     while(n!=0) {
11         n2=pow(n,2);
12         n3=pow(n,3);
13         cout << "Your number is: " <<n
14         <<" To tetragono tou ine: "<<n2
15         <<" ke o kivos toyu ine: "<<n3<<endl;
16         cout<<"insert a new number "<<endl;
17         cin>>n;
18     }
19     cout<< "End of program";
20     return 0;
21 }
```

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main()
6  {
7      int n, n2=0,n3=0;
8      do{
9          cout<<"insert a number "<<endl;
10         cin>>n;
11         n2=pow(n,2);
12         n3=pow(n,3);
13         cout << "Your number is: " <<n
14         <<" To tetragono tou ine: "<<n2
15         <<" ke o kivos toyu ine: "<<n3<<endl;
16     }
17     while(n!=0);
18     cout<< "End of program";
19     return 0;
20 }
```

```
insert a number
2
Your number is: 2 To tetragono tou ine: 4 ke o kivos toyu ine: 8
insert a new number
6
Your number is: 6 To tetragono tou ine: 36 ke o kivos toyu ine: 216
insert a new number
0
End of program
Process returned 0 (0x0)   execution time : 14.471 s
Press any key to continue
```

# Έλεγχος δεδομένων εισόδου

33

- Να δημιουργήσετε ένα πρόγραμμα που θα δέχεται συγκεκριμένο πλήθος αριθμών. Το πλήθος καθορίζεται από τον χρήστη και πρέπει να είναι μεγαλύτερο του 1 και μικρότερο του 100. Το πρόγραμμα θα υπολογίζει και εμφανίζει τον μικρότερο, τον μεγαλύτερο, το άθροισμα και τον μέσο όρο όλων των αριθμών που εισήχθησαν.
- Να δώσετε μια λύση χρησιμοποιώντας την εντολή `while` και μια δεύτερη λύση χρησιμοποιώντας την εντολή `do - while`

# 'Ελεγχος δεδομένων εισόδου exed03

34

```
Dose Arithmo K:8  
Number=[1]=3  
Number=[2]=4  
Number=[3]=6  
Number=[4]=3  
Number=[5]=7  
Number=[6]=1  
Number=[7]=11  
Number=[8]=6  
To Minimum=1  
To Maximum=11  
O Mesos Oros=5.125  
to athroisma=41
```

```
1  #include <iostream>  
2  using namespace std;  
3  int main() {  
4      int i,k;  
5      float sum, num, avg, minimum, maximum;  
6      sum=0;  
7      minimum=1000;  
8      maximum=0;  
9      cout<<"Dose Arithmo K:";  
10     cin>>k;  
11     while (k<0 || k>=100){  
12         cout<<"ERROR"<<endl;  
13         cout<<"Dose K Ksana:";  
14         cin>>k;  
15     }  
16     for (i=1; i<=k; i++){  
17         cout<<"Number=["<<i<<"]=";  
18         cin>>num;  
19         if (minimum>num)  
20             minimum=num;  
21         if (maximum<num)  
22             maximum=num;  
23         sum+=num;  
24     }  
25     avg=sum/k;  
26     cout<<"To Minimum="<<minimum<<endl;  
27     cout<<"To Maximum="<<maximum<<endl;  
28     cout<<"O Mesos Oros="<<avg<<endl;  
29     cout<<"to athroisma="<<sum<<endl;  
30     return 0;  
31 }  
32
```

```
1  #include <iostream>  
2  
3  using namespace std;  
4  
5  int main()  
6  {  
7      int i,k;  
8      float num, sum=0, avg, maxim=0, minim=1000;  
9      do {  
10         cout<<"Who many numbers?(between 1 and 100) "<<endl;  
11         cin>>k;  
12     }  
13     while(k<1 || k>100);  
14     for(i=1;i<=k;i++){  
15         cout<<"insert the "<<i<<" number "<<endl;  
16         cin>>num;  
17         if(minim>num) minim=num;  
18         if(maxim<num) maxim=num;  
19         sum+=num;  
20     }  
21     avg=sum/k;  
22     cout<<"Minimum : "<<minim<<endl;  
23     cout<<"Maximum : "<<maxim<<endl;  
24     cout<<"Athrisma : "<<sum<<endl;  
25     cout<<"Mesos Oros : "<<avg<<endl;  
26     cout << "Hello world!" << endl;  
27     return 0;  
28 }
```

# Παράδειγμα 2 do pxedd02

35

- Να δημιουργήσετε πρόγραμμα το οποίο να δέχεται δύο ακέραιους αριθμούς και να εμφανίζει στην οθόνη τις επιλογές:

1. Addition,
2. Multiplication,
3. Subtraction,
4. Division,
5. Modulo,
6. Exit.

Ανάλογα με την επιλογή του χρήστη, να εμφανίζεται το αποτέλεσμα της αντίστοιχης αριθμητικής πράξης.

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int op,a,b;
5     cout << "Enter two numbers (a>b): ";
6     cin >> a >> b;
7     do {
8         cout << "1. Addition" << endl;
9         cout << "2. Multiplication" << endl;
10        cout << "3. Subtraction" << endl;
11        cout << "4. Division" << endl;
12        cout << "5. Modulo" << endl;
13        cout << "6. Exit" << endl;
14        cout << "Please make a choice: ";
15        cin >> op;
16
17        switch(op){
18            case 1:cout << a << "+" << b << "=" <<a+b << endl;
19                break;
20            case 2:cout << a << "*" << b << "=" <<a*b<< endl;
21                break;
22            case 3:cout << a << "-" << b << "=" <<a-b<< endl;
23                break;
24            case 4:cout << a << "/" << b << "=" <<a/b<< endl;
25                break;
26            case 5:cout << a << "%" << b << "=" <<a%b<< endl;
27                break;
28            case 6:cout << "Program ended"<< endl;
29                break;
30            default:cout << "Wrong input" << endl;
31        }
32    }
33    while(op!=6);
34    return 0;
35 }
36
```

# Άσκηση For ή While ή Do While exed01

36

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα σε C++ που να δέχεται ένα ακέραιο αριθμό μικρότερο από 100 και να υπολογίζει το άθροισμα όλων των ακέραιων αριθμών από το 1 μέχρι τον αριθμό αυτό. Πχ αν δοθεί ο αριθμός 5 να υπολογίζει το άθροισμα  $1+2+3+4+5 = 15$ . Να δώσετε μια λύση με την χρήση βρόχου for, While ή Do While προσθέτοντας την δυνατότητα ελέγχου των δεδομένων εισόδου

```
1  #include <iostream>
2
3  using namespace std;
4  // Find 1+2+3+....n
5  int main()
6  {
7      int n, sum = 0;
8      do {
9          cout << "\n Enter a number from 1 to 100.: " << endl;
10         cin>>n;
11         while(n<1||n>100);
12         for(int i=1;i<=n;i++) {
13             sum = sum + i;
14         }
15         cout<<"\n The sum from 1 to " <<n<<"\n is: " <<sum<<endl;
16         return 0;
17     }
```

# Break / Continue exic0a & exic0b

37

Χρησιμοποιούνται για τον καλύτερο έλεγχο της ροής ενός προγράμματος

- Break – για να σταματήσει η εκτέλεση των εντολών του βρόχου.
- Continue - για να υπερπηδήσει μια τιμή

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int counter=0;
7      while (counter < 10) {
8          cout << counter << endl;
9          counter++;
10         if (counter==5){
11             cout << "Reached 5, now exiting!"
12             <<endl;
13             break;
14         }
15     }
16     return 0;
17 }
18
```

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      for (int n=10; n>0; n--){
7          if (n==5) continue;
8          // number 5 is skipped
9          cout << n << ", ";
10         }
11     return 0;
12 }
13
```

# Άσκηση brake exbc01

38

Θέλουμε να δημιουργήσουμε ένα πρόγραμμα σε C++ που να δέχεται μέχρι 50 ακέραιους αριθμούς μικρότερους από 100. Να γίνεται έλεγχος δεδομένων και να υπολογίζει το εμβαδό και την περίμετρο κύκλου με ακτίνα τον αριθμό που δόθηκε. Αν το εμβαδό είναι μεγαλύτερο από το διπλάσιο της περιμέτρου να σταματά η εκτέλεση του προγράμματος.

$$E=3,14R^2$$

$$P = 2*3,14*R$$

ΛΥΣΗ...

```
16         perimetros=2*3.14*rad;
17         if(embvado>2*perimetros)
18             cout<<"To embvado is grater than Perimetros x 2"<<endl;
19             break;
20     }
21     cout<<"the " <<i<< " st number is " <<rad;
22         <<" The embvado = " <<embvado<<" Perimetros x 2 is " <<perimetros*2<<endl;
23 }
24 cout<<"Telos programmatos"<<endl;
25 return 0;
26 }
```

# Break exed02

39

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να δέχεται ακέραιους αριθμούς και να παρουσιάζει στην οθόνη το τετράγωνο και τον κύβο τους. Το πρόγραμμα θα σταματήσει να δέχεται αριθμούς, όταν δοθεί ο αριθμός 0.

ΛΥΣΗ...

```
13     n2=pow(n,2);
14     n3=pow(n,3);
15     cout << "Your number is: " <<n
16     <<" To tetragono tou ine: " <<n2
17     <<" ke o kivros tou ine: " <<n3<<endl;
18 }
19 cout<< "End of program";
20 return 0;
21 }
```

# Φωλιασμένοι βρόχοι (Nested loops) exed2f01

- Ποιο πιστεύετε θα είναι το αποτέλεσμα του πιο κάτω παραδείγματος;

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      for (char letter='a';letter<'l';letter++){
8          cout << letter << endl;
9          for(int i=1;i<11;i++){
10             cout << i;
11         }
12         cout << endl;
13     }
14     return 0;
15 }
16
```

```
a
12345678910
b
12345678910
c
12345678910
d
12345678910
e
12345678910
f
12345678910
g
12345678910
h
12345678910
i
12345678910
j
12345678910
k
12345678910
```

# Φωλιασμένοι βρόχοι (Nested loops) exed2f02

- Ποιο πιστεύετε θα είναι το αποτέλεσμα του πιο κάτω παραδείγματος;

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int row;
7      char col;
8      for (row = 0; row < 5; row++) {
9          cout << "\n" << row;
10         for (char col = 'a'; col < 'd'; col++){
11             cout << "\t" << col << row;
12         }
13         cout << "\t**";
14     }
15
16 }
17
```

```
0      a0      b0      c0      **
1      a1      b1      c1      **
2      a2      b2      c2      **
3      a3      b3      c3      **
4      a4      b4      c4      **
Process returned 0 (0x0)   execution time : 0.547 s
Press any key to continue.
```

# Φωλιασμένοι βρόχοι (Nested loops) exed2f03

42

Μια γραμμή αστικού λεωφορείου έχει την πρώτη αναχώρηση της ημέρας από την αφετηρία στις 6 π.μ. και νέα αναχώρηση κάθε 15 λεπτά. Το τελευταίο δρομολόγιο της μέρας αναχωρεί η ώρα 6 μ.μ. Να δημιουργήσετε ένα πρόγραμμα σε C++ που να δέχεται την ώρα έναρξης του πρώτου και τελευταίου δρομολογίου, τον χρόνο ανάμεσα σε δυο συνεχόμενα δρομολόγια, και θα τυπώνει την ώρα αναχώρησης όλων των δρομολογίων της ημέρας. Σημ. Έχουμε ακέραιο αριθμό δρομολογίων ανά ώρα.

Άριστος Πασιάς

```
Insert the time of the first departure
6
Insert the time of the Last departure
9
Insert the time between two departures
20
Dromologio Imeras arithomos 1 6:0
Dromologio Imeras arithomos 2 6:20
Dromologio Imeras arithomos 3 6:40
Dromologio Imeras arithomos 4 7:0
Dromologio Imeras arithomos 5 7:20
Dromologio Imeras arithomos 6 7:40
Dromologio Imeras arithomos 7 8:0
Dromologio Imeras arithomos 8 8:20
Dromologio Imeras arithomos 9 8:40
```

# Φωλιασμένοι βρόχοι (Nested loops) exed2f03

43

ΛΥΣΗ...

```
13     cin>>step;
14     for(int i=d0;i<d1;i++){
15         for(int j=0;j<60;j=j+step){
16             n=n+1;
17             cout<<"Dromologio Imeras arithmos "<<n<<"\t"<<i<<":"<<j<<endl;
18         }
19     }
20     return 0;
21 }
```

# Φωλιασμένοι βρόχοι (Nested loops) exed2f04

44

- Να δημιουργήσετε μια εφαρμογή σε C++ η οποία να ρωτά τον αριθμό γραμμών και να σχεδιάζει το πιο κάτω σχήμα.

```
Enter number of rows: 6
 *
 ***
 *****
 ****
 *****
 *****
 *****
 *****

Process returned 0 (0x0)   execution time: 0:00.000
Press any key to continue.
```



# Φωλιασμένοι βρόχοι (Nested loops) exed2f05

45

- Να δημιουργήσετε μια εφαρμογή σε C++ η οποία να ρωτά τον αριθμό γραμμών και να σχεδιάζει το πιο κάτω σχήμα.

```
Enter number of rows: 8
```

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

```
Process returned 0 (0x0)   exe
Press any key to continue.
```



Try it out

# Φωλιασμένοι βρόχοι (Nested loops) exed2f06

46

- Να δημιουργήσετε μια εφαρμογή σε C++ η οποία θα εμφανίζει όλους τους πρώτους αριθμούς μεταξύ του 1 και του 100.

```
2 is prime  
3 is prime  
5 is prime  
7 is prime  
11 is prime  
13 is prime  
17 is prime  
19 is prime  
23 is prime  
29 is prime  
31 is prime  
37 is prime  
41 is prime  
43 is prime  
47 is prime  
53 is prime  
59 is prime  
61 is prime  
67 is prime  
71 is prime  
73 is prime  
79 is prime  
83 is prime  
89 is prime  
97 is prime
```



# Ασκήσεις σελίδα 162 βιβλίου

47

## Ασκήσεις:

4.1

4.3

4.4

4.23

4.8

4.17



## Άσκηση 4.1

- Ποια από τις πιο κάτω δομές επανάληψης τυπώνει το σύμβολο «\*» πέντε φορές στην οθόνη;

(α) `for (i=8; i>=3; i--)`

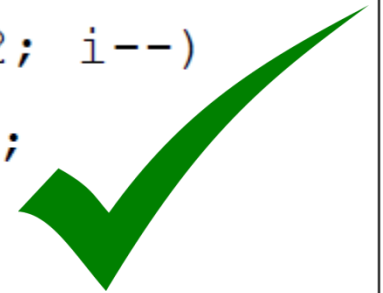
6

`cout << "*";`

(β) `for (i=6; i>=2; i--)`

5

`cout << "*";`



(γ) `for (i=10; i<21; i+=2)`

6

`cout << "*";`

(δ) `for (i=0; i<=6; i++)`

7

`cout << "*";`

## Άσκηση 4.3

- Ποιο θα είναι το αποτέλεσμα, μετά την εκτέλεση του πιο κάτω κώδικα;

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int i, a;
8      for (i=5; i<=10; i++){
9          a = 10 * i;
10         cout << a << " " << i << endl;
11     }
12     return 0;
13 }
14
```

```
50 5
60 6
70 7
80 8
90 9
100 10
```



# Άσκηση 4.4

50

- Δίνεται το πρόγραμμα που δέχεται N ακέραιους αριθμούς και παρουσιάζει το πλήθος των θετικών, των αρνητικών και των μηδενικών αριθμών. Να γράψετε τους ελέγχους που πρέπει να γίνουν, ώστε να ελέγξετε την ορθή λειτουργία του προγράμματος και να παρουσιάσετε για έναν έλεγχο τον πίνακα της προκαταρκτικής εκτέλεσης.

```
1 #include<iostream>
2 using namespace std;
3 int main(){
4     int pos=0,neg=0,zeros=0,i,N,num;
5     cin>>N;
6     for(i=1;i<=N;i++){
7         cin>>num;
8         if(num>0)
9             pos+=1;
10        else if(num<0)
11            neg+=1;
12        else
13            zeros+=1;
14    }
15    cout<<pos<<endl;
16    cout<<neg<<endl;
17    cout<<zeros<<endl;
18    return 0;
19 }
20
```

Τιμή N	Τιμές εισόδου	Αρνητικοί	0	Θετικοί
N=5	-1 -100, 0, 1, 100	2	1	2
N=-1	0	0	0	0
N=3	-1, -1, -1	3	0	0
N=3	0,0,0	0	3	0
N=3	2,2,2,	0	0	3

# Άσκηση 4.23

51

Το πιο κάτω πρόγραμμα δέχεται δέκα ακέραιους αριθμούς και τυπώνει τον μεγαλύτερο αριθμό και τη σειρά στην οποία δόθηκε. Σε περίπτωση που υπάρχουν περισσότεροι από έναν μέγιστοι αριθμοί, τυπώνεται ο τελευταίος που δόθηκε. Στο πρόγραμμα υπάρχουν πέντε λογικά ή/και συντακτικά λάθη/παραλείψεις. Να ξαναγράψετε το πρόγραμμα, διορθώνοντας τα λάθη/παραλείψεις.

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int num, max, maxth;
5      cin>>num;
6      max=num;
7      for (i=1;i<=10;i++){
8          cin>>num;
9          if (num>max) {
10             max=num;
11             maxth=num;
12         }
13     }
14     cout<<"Max="<<max<<" thesi="<<maxth;
15     return 0;
16 }
```

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int num, max, maxth;
5      cin>>num;
6      max=num;
7      maxth=1; //
8      for (int i=2;i<=10;i++){ //int i=2
9          cin>>num;
10         if (num>=max) { //|=
11             max=num;
12             maxth=i; //not maxth=num
13         }
14     }
15     cout<<"Max="<<max<<" thesi="<<maxth;
16     return 0;
17 }
18
```

## Άσκηση 4.8

- Να δημιουργήσετε πρόγραμμα το οποίο να ζητά έναν ακέραιο θετικό αριθμό  $N \leq 100$  και να βρίσκει και να εμφανίζει στην οθόνη το άθροισμα όλων των αριθμών από το 1 μέχρι το  $N$ . Για παράδειγμα, αν ο χρήστης δώσει τον αριθμό 10, το πρόγραμμα θα βρίσκει το άθροισμα:  
 $S=1+2+3+4+5+6+7+8+9+10 = 55$



```
insert an integer10
```

```
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10 + = 55
```

```
Process returned 0 (0x0)   execution time : 3.437 s
```

```
Press any key to continue.
```

Try it out

# Άσκηση 4.17

53

- Ένα θέατρο διαθέτει 30 σειρές καθισμάτων. Η πρώτη σειρά διαθέτει 50 καθίσματα και κάθε επόμενη σειρά διαθέτει 10 περισσότερα από την προηγούμενη. Να δημιουργήσετε πρόγραμμα το οποίο να υπολογίζει και να εμφανίζει τον αριθμό των καθισμάτων της τελευταίας σειράς, καθώς και τη συνολική χωρητικότητα καθισμάτων του θεάτρου.

```
last row has 340 seats and the total capacity is 5800  
Process returned 0 (0x0)   execution time : 0.673 s  
Press any key to continue.
```



# Γενικές Ασκήσεις

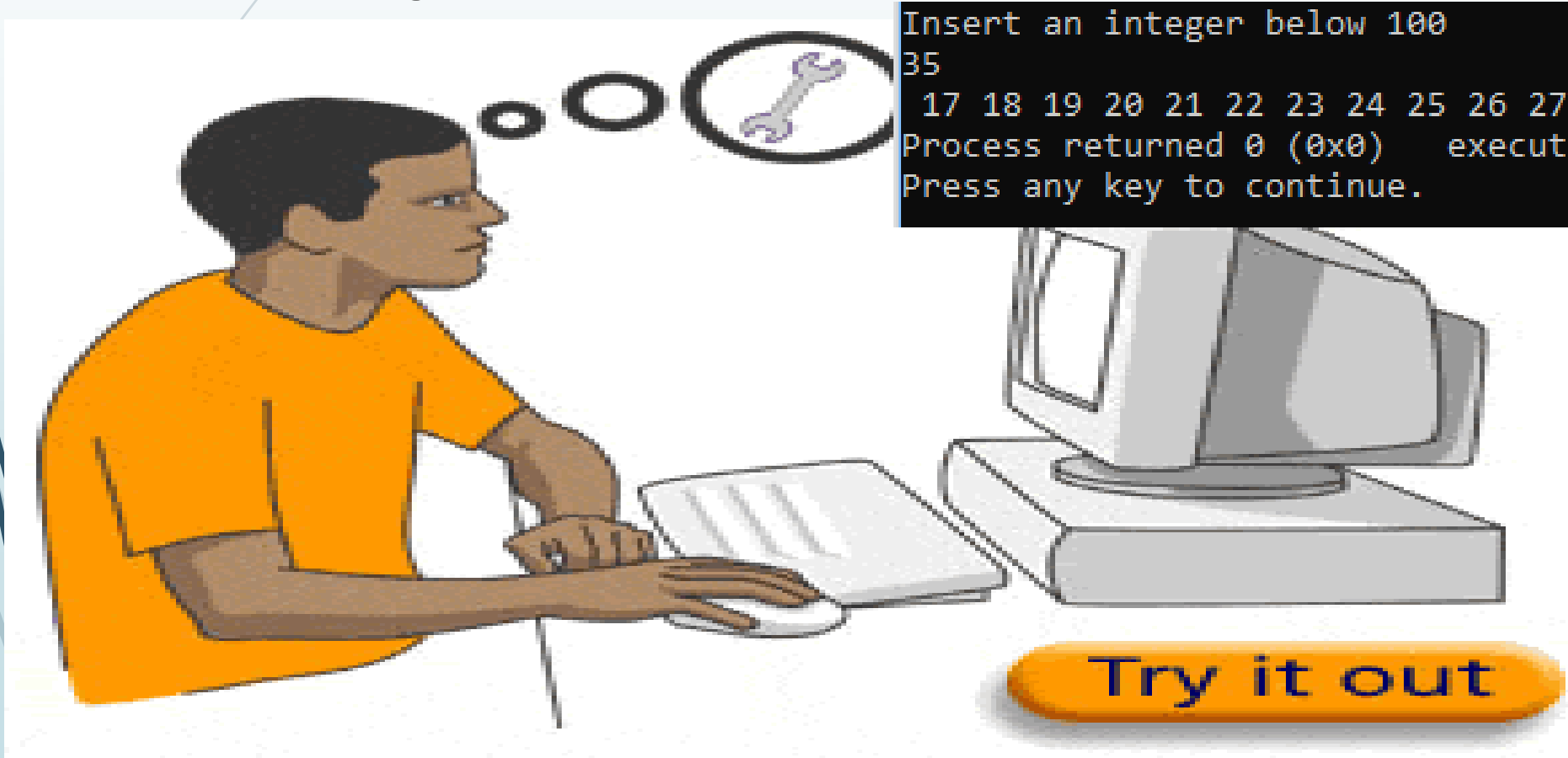
54



# Άσκηση 1 (exg01)

55

Να δημιουργήσετε πρόγραμμα σε C++ το οποίο να δέχεται ένα ακέραιο θετικό αριθμό μικρότερο του 100. Αφού γίνει έλεγχος των δεδομένων εισόδου στη συνέχεια να τυπώνει όλους τους ακέραιους αριθμούς που είναι μικρότεροι από N και μεγαλύτεροι από N/2.



```
Insert an integer below 100
35
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34
Process returned 0 (0x0)   execution time : 6.581 s
Press any key to continue.
```

Να δημιουργήσετε πρόγραμμα το οποίο να διαβάζει από τη γραμμή εντολής ένα ακέραιο θετικό αριθμό  $N$  μικρότερο από το 20 και να τυπώνει την ακολουθία των αθροισμάτων από το 1 μέχρι το  $i$ , για κάθε  $i$  από το 1 μέχρι το  $N$ . Για παράδειγμα για  $N=10$ , το αποτέλεσμα του προγράμματος είναι το ακόλουθο: 1, 3, 6, 10, 15, 21, 28, 36, 45, 55 **Να γίνεται έλεγχος των δεδομένων εισόδου**



```
Insert a number between 1 and 20.  
25  
Insert a number between 1 and 20.  
-7  
Insert a number between 1 and 20.  
10  
1 3 6 10 15 21 28 36 45 55
```

Να γράψετε πρόγραμμα στην C++ που να διαβάζει από τη γραμμή εντολών ακέραιους θετικούς αριθμούς μικρότερους του 100. **Να κάνει έλεγχο δεδομένων εισόδου** και να εμφανίζει συνέχεια το μέσο όρο των αριθμών αυτών. Το πρόγραμμα να σταματά όταν ο μέσος όρος πάρει τιμή η οποία είναι ακέραιο πολλαπλάσιο του 5



# Άσκηση 4 (exg04)

58

- Να σχεδιάσετε ένα λογικό διάγραμμα και να δημιουργήσετε το αντίστοιχο πρόγραμμα το οποίο να υπολογίζει και να παρουσιάζει τον πληθυσμό της Κύπρου για τα επόμενα 20 χρόνια, αν ο πληθυσμός της αυτή τη στιγμή είναι 800 000 και η ετήσια αύξηση είναι 1% ανά έτος.
- **Σημείωση**. Θα πρέπει να παρουσιάζει τον πληθυσμό και την αύξηση κάθε χρονιά ξεχωριστά όπως στο πιο κάτω παράδειγμα.

Χρονos	Plithismos	Αuksisi
0	800000	0
1	808000	8000
2	816080	8080
3	824241	8161
4	832483	8242

# Άσκηση 4 Λύση

59



xronos	plithismos	Afxisi
0	800000	0
1	808000	8000
2	816080	8080
3	824241	8160.8
4	832483	8242.41
5	840808	8324.83
6	849216	8408.08
7	857708	8492.16
8	866285	8577.08
9	874948	8662.85
10	883698	8749.48
11	892535	8836.98
12	901460	8925.35
13	910475	9014.6
14	919580	9104.75
15	928775	9195.79
16	938063	9287.75
17	947444	9380.63
18	956918	9474.44
19	966487	9569.18
20	976152	9664.87

Τ Ε Λ Ο Σ !

# Άσκηση 5 (exg05)

60

- Να δημιουργήσετε ένα πρόγραμμα το οποίο να υπολογίζει και να παρουσιάζει τον πληθυσμό της Κύπρου για τα επόμενα χρόνια μέχρι να ξεπεράσει το 1000000, αν ο πληθυσμός της αυτή τη στιγμή είναι 800000 και η ετήσια αύξηση είναι 2%.
- **Σημείωση**. Θα πρέπει να παρουσιάζει τον πληθυσμό και την αύξηση κάθε χρονιά ξεχωριστά όπως στο πιο κάτω παράδειγμα.

xronos	plithismos	Afxisi
0	800000	0
1	816000	16000
2	832320	16320
3	848966	16646.4
4	865946	16979.3
5	883265	17318.9
6	900930	17665.3
7	918949	18018.6
8	937328	18379
9	956074	18746.6
10	975196	19121.5
11	994700	19503.9

T E L O S !

# Άσκηση 5 Λύση

61

xronos	plithismos	Afxisi
0	800000	0
1	816000	16000
2	832320	16320
3	848966	16646.4
4	865946	16979.3
5	883265	17318.9
6	900930	17665.3
7	918949	18018.6
8	937328	18379
9	956074	18746.6
10	975196	19121.5
11	994700	19503.9

S !



# Άσκηση 6

62

- Σε ένα εστιατόριο κάθε πελάτης πληρώνει επιπλέον 5% της αξίας του λογαριασμού, για φιλανθρωπικούς σκοπούς. Να δημιουργήσετε πρόγραμμα το οποίο λαμβάνει από τη γραμμή εντολής ένα απροσδιόριστο πλήθος ποσών λογαριασμών και υπολογίζει το συνολικό ποσό που θα δοθεί για φιλανθρωπικούς σκοπούς.

# Άσκηση 7

63

- Να δημιουργήσετε πρόγραμμα το οποίο λαμβάνει από τη γραμμή εντολής ένα ακέραιο θετικό αριθμό  $N$  που αντιπροσωπεύει το βάρος σε κιλά μιας στοίβας από άμμο, και υπολογίζει το ελάχιστο πλήθος κουβάδων χωρητικότητας 5 κιλών, 3 κιλών και ενός κιλού που χρειάζονται για να μαζέψουν την άμμο.

# Άσκηση 8

64

Να δημιουργήσετε πρόγραμμα το οποίο να ζητά από τον χρήστη αριθμούς από το 1 μέχρι το 9999. Αν ο αριθμός είναι εκτός ορίων, τότε το πρόγραμμα να τυπώνει το μήνυμα «Wrong input» και να ζητά άλλο αριθμό. Για να τερματιστεί το πρόγραμμα, ο χρήστης θα πρέπει να δώσει τον αριθμό 0 (μηδέν). Το πρόγραμμα θα υπολογίζει το πλήθος των μονοψήφιων, των διψήφιων, των τριψήφιων και των τετραψήφιων αριθμών που έχουν δοθεί.

# Άσκηση 9

65

- Στην Ολυμπιάδα Πληροφορικής λαμβάνουν μέρος οι μαθητές του Γ' έτους. Να σχεδιάσετε λογικό διάγραμμα και να δημιουργήσετε πρόγραμμα το οποίο:
  - να δέχεται από τον χρήστη τον αριθμό των διαγωνιζομένων, το όνομα και την βαθμολογία κάθε διαγωνιζόμενου. Αν η βαθμολογία δεν είναι μεταξύ 1 και 100 συμπεριλαμβανομένων, να ζητά από τον χρήστη να την ξαναδώσει
  - να υπολογίζει τον μέσο όρο των βαθμών όλων των διαγωνιζομένων
  - να βρίσκει το πλήθος των μαθητών με βαθμό πάνω από 50
  - να βρίσκει τη μεγαλύτερη βαθμολογία και να τυπώνει το όνομα του μαθητή που την πέτυχε

# Άσκηση 10

66

- Ένα κοσμηματοπωλείο πωλεί διάφορα κοσμήματα στα οποία ο πωλητής έχει δικαίωμα μιας μέγιστης έκπτωσης ανάλογα με το κόσμημα. Σε κάθε κόσμημα υπάρχει μια καρτέλα στην οποία αναγράφεται ένας αριθμός μητρώου που είναι μοναδικός. Ο αριθμός μητρώου είναι επταψήφιος ακέραιος αριθμός. Το γινόμενο των δύο τελευταίων ψηφίων του αριθμού μητρώου καθορίζει την μέγιστη έκπτωση για το συγκεκριμένο κόσμημα. Να δημιουργήσετε πρόγραμμα το οποίο να δέχεται τον αριθμό μητρώου και την κανονική τιμή ενός κοσμήματος και να υπολογίζει και να τυπώνει τον αριθμό μητρώου, την αρχική τιμή, την έκπτωση και την τιμή μετά την έκπτωση.

# Ασκήσεις σελίδα 168 βιβλίου

67

**Ασκήσεις:**

**4.25**

**4.27**

**4.28**

**4.30**

