

03 – Εντολές επανάληψης



Στα προβλήματα που επιλύονται με υπολογιστή συχνά χρειάζεται να επαναληφθεί η εκτέλεση μιας σειράς εντολών. Μία εντολή επανάληψης ορίζει ένα τμήμα/block κώδικα το οποίο εκτελείται κατά επανάληψη με βάση την τιμή (true ή false) μίας συνθήκης (condition).

Οι εντολές που συγκροτούν μια εντολή επανάληψης αποκαλούνται **βρόχος** (loop).

Η C++ διαθέτει τις εντολές επανάληψης **while**, **for**, **do**.

Εντολή **while**

```
while ( <condition> )
{
    //block κώδικα που εκτελείται επαναληπτικά όσο η συνθήκη είναι true
}
```

- Ⓢ Η συνθήκη (condition) είναι μια λογική έκφραση και λαμβάνει την τιμή true ή false.
- Ⓢ Θα πρέπει να εξασφαλίσετε ότι κάποια στιγμή η συνθήκη θα γίνει false προκειμένου το πρόγραμμά σας να μην εγκλωβιστεί και επαναλαμβάνει το ίδιο κομμάτι κώδικα για πάντα (**ατέρμων βρόχος**).
- Ⓢ Παρατηρήστε ότι υπάρχει η περίπτωση η συνθήκη στον πρώτο έλεγχο της να είναι false και τότε δεν θα εκτελεστούν ποτέ οι εντολές μέσα στο block { } της εντολής while.
- Ⓢ Εάν έχετε ένα τμήμα κώδικα που δεν γνωρίζετε πόσες φορές πρέπει να επαναληφθεί η εκτέλεσή του πρέπει να χρησιμοποιήσετε την εντολή **while** ή **do** (όχι την εντολή **for**).

Παράδειγμα 1: Το ακόλουθο πρόγραμμα εμφανίζει στην οθόνη όλους τους αριθμούς από το 0 μέχρι και το 9. Παρατηρήστε πως ελέγχεται κατά επανάληψη η συνθήκη $i < 10$ μέχρι να γίνει false και να τερματίσει η εκτέλεση του βρόχου.

```
#include <iostream> // 03_example1.cpp
using namespace std;
1 int main ()
2 {
3     int i = 0;
4     while ( i < 10 )           // Όσο η i είναι μικρότερη του 10
5     {
6         cout << i << "\n";    // εμφάνισε την τιμή της i
7         i++;                  // αύξησε την i κατά μία μονάδα (ή i=i+1;)
8     }                         //επιστροφή στην γραμμή 4 while για τον επόμενο έλεγχο της συνθήκης
9 }
```

Άσκηση Α

A1. Τι αλλαγές πρέπει να γίνουν στο παραπάνω πρόγραμμα ώστε να εμφανίζει τους αριθμούς από το 1 μέχρι και το 20;

A2. Τι αλλαγές πρέπει να γίνουν στο παραπάνω πρόγραμμα ώστε να εμφανίζει τους αριθμούς 2, 4, 6, 8, 10, 12 και 14 ;

A3. Γιατί είναι απαραίτητη η εντολή `i++`; (ή `i=i+1`; ή `i=i+2`;) στο εσωτερικό του block;

Εντολή `for`

```
for (variable initialization; condition; variable update)
{
    // block κώδικα που εκτελείται επαναληπτικά όσο η συνθήκη είναι true
}
```

- ⊗ Παρατηρείστε την χρήση του `;` μεταξύ των παραμέτρων μέσα στην παρένθεση.
- ⊗ Στη θέση του *variable initialization* δίνεται η αρχική τιμή στην μεταβλητή που ελέγχει την επανάληψη (π.χ. `int i=0`).
- ⊗ Στη θέση του *condition* γράφεται η συνθήκη που όσο είναι `true` θα επαναλαμβάνεται η εκτέλεση του βρόχου. (πχ `i<10`).
- ⊗ Στη θέση του *variable update* γράφεται η εντολή που ορίζει την μεταβολή στην μεταβλητή που ελέγχει την επανάληψη. Για παράδειγμα θα μπορούσαμε να συναντήσουμε την εντολή `i++` η οποία ορίζει πως η `i` αυξάνεται κατά μία μονάδα (βήμα 1) ή την εντολή `i=i+5` η οποία ορίζει πως η `i` αυξάνεται κατά 5 μονάδες σε κάθε νέα επανάληψη (βήμα 5).
- ⊗ Το *variable update*, δηλαδή η μεταβολή της μεταβλητής που ελέγχει την επανάληψη πραγματοποιείται κάθε φορά στο **τέλος** της εκτέλεσης του block `{ }` της `for`.
- ⊗ Παρατηρείστε ότι υπάρχει η περίπτωση η συνθήκη να είναι `false` στον πρώτο έλεγχο και τότε δεν θα εκτελεστούν καθόλου οι εντολές της `for`.
- ⊗ Εάν γνωρίζετε εκ των προτέρων το πλήθος των επαναλήψεων, τότε προτιμήστε την εντολή `for`.

Παράδειγμα

```
for (int i=0; i < 10; i++)
{
    cout << i << "\n";
} //ορίζει το τέλος της for
```

Βήματα εκτέλεσης

- Δίνεται αρχική τιμή 0 στην μεταβλητή `i` (variable initialization: `int i=0`;))
- Ελέγχεται η συνθήκη `i < 10` (condition: `i < 10`) η οποία είναι `true` και άρα εκτελεί τον βρόχο της `for` και εμφανίζεται η τιμή 0
- Προστίθεται μία μονάδα στην `i` (variable update: `i++`) και παίρνει τιμή 1.
- Ελέγχεται η συνθήκη `i < 10` η οποία είναι `true` και άρα εκτελεί το block της `for` και εμφανίζεται η τιμή 1
- Προστίθεται μία μονάδα στην `i` (variable update `i++`) και παίρνει τιμή 2.
- Ελέγχεται η συνθήκη `i < 10` η οποία είναι `true` και άρα εκτελεί το block της `for` και εμφανίζεται η τιμή 2
- .. επανάληψη των προηγούμενων βημάτων, κάθε φορά με προσαυξημένη τιμή του `i` κατά 1

- ..
- εμφανίζεται η τιμή 9
- Προστίθεται μία μονάδα στην i (variable update i++) και παίρνει τιμή 10.
- Ελέγχεται η συνθήκη $i < 10$ η οποία είναι false και τερματίζεται ο βρόχος της for.
- Εκτελείται η εντολή που ακολουθεί το κλείσιμο } του block της for .

Παρατήρηση: Το παράδειγμα με *while* και το παραπάνω παράδειγμα με *for* έχουν ακριβώς την ίδια λειτουργία. Η υλοποίηση με εντολή *for* είναι ωστόσο πιο συμπαγής.

Άσκηση Β

Γράψτε ένα πρόγραμμα που θα ζητά από τον χρήστη να εισάγει τον κωδικό του. Το πρόγραμμα θα πρέπει να συνεχίζει να ζητά τον κωδικό μέχρι να εισαχθεί ο σωστός – για την άσκησή μας ο κωδικός θα είναι *Pierce*. Όταν εισαχθεί σωστός κωδικός το πρόγραμμα να εμφανίζει το μήνυμα «You made attempts to type the correct password» Σκεφτείτε αν γνωρίζετε από πριν πόσες φορές θα χρειαστεί να εισάγει ο χρήστης τον κωδικό του.

- Ποια εντολή επανάληψης είναι κατάλληλη και γιατί; _____

Εντολές Break και Continue

- Η εντολή **break** μέσα στον βρόχο σταματά την εκτέλεση του βρόχου και τερματίζει την επανάληψη .
- Η εντολή **continue** σταματά την εκτέλεση του βρόχου και συνεχίζει με την νέα τιμή της μεταβλητής ελέγχου.

Εντολή do-while

```
do {
    //block κώδικα που εκτελείται επαναληπτικά
} while (<condition>;
```

Εκτελεί πάντα μία φορά τις εντολές μέσα στο block του κώδικα και έπειτα ελέγχει την συνθήκη. Όσο η συνθήκη είναι true επαναλαμβάνεται η εκτέλεση του block του κώδικα. Η επανάληψη τερματίζει όταν η συνθήκη είναι false.

Το παράδειγμα που ακολουθεί εμφανίζει όποια τιμή εισάγει ο χρήστης μέχρι να εισάγει την τιμή goodbye:

```
#include <iostream>
#include <string>
using namespace std;
int main ()
{
    string str;
    do {
        cout << "Enter text: ";
        cin>>str;
        cout << "You entered: "<<str<<'\n';
    } while (str != "goodbye");
}
```

Enter text: hello
 You entered: hello
 Enter text: Anyone?
 You entered: Anyone?
 Enter text: goodbye
 You entered: goodbye

Προτιμάται η χρήση της εντολής *do* όταν απαιτείται να εκτελεστεί ο βρόχος της επανάληψης τουλάχιστον μία φορά. Για παράδειγμα, όταν το πρόγραμμα πρέπει να δεχθεί έγκυρη τιμή

είσοδου (πχ να δέχεται μόνο τις τιμές 1, 2 ή 3).

```
do {
    cout << "Enter choice 1 or 2 or 3 only: ";
    cin>>num;
} while (num!=1 && num!=2 && num!=3);
```

Αυτοαξιολόγηση

1) Ποια θα είναι η τιμή της μεταβλητής x μετά την ολοκλήρωση της εκτέλεσης της εντολής:

for (int x=0; x<20; x++) {...}

a) 20

c) 0

b) 19

d) 1

2) Πότε θα εκτελείται το block του κώδικα που ακολουθεί την while(x<100);

a) Όσο το x είναι μικρότερο από το 10

d) Εξαρτάται από τις εντολές του block

b) Όσο το x είναι μικρότερο από το 100

της while

c) Όσο το x είναι ίσο με το 100

3) (α) Εκτελέσετε εικονικά τον παρακάτω κώδικα και γράψτε τι θα εμφανίσει ως έξοδο.

Έστω πως εισάγεται η τιμή 5 στην μεταβλητή stop.

```
#include <iostream>
using namespace std;
int main ()
{
    int sum, oros, stop;
    cin>>stop;
    oros=1;
    sum=0;
    while (sum<stop)
    {
        sum=sum+oros;
        oros=oros+1;
        cout<<"the sum now is "<<sum<<"\n";
    }
    cout <<" end of loop \n";
    return 0;
}
```


(β) Θα μπορούσε να υλοποιηθεί το πρόγραμμα με την εντολή for αντί της while;

Αιτιολογήστε την απάντησή σας. _____

(γ) Τι υπολογίζεται στην μεταβλητή sum; _____

4. (α) Δίνεται το παρακάτω πρόγραμμα το οποίο θα εκτελέσει last πλήθος επαναλήψεων και θα υπολογίσει το άθροισμα 1+2+....+last. Να εντοπίσετε ποια είναι η μεταβλητή που ελέγχει τις επαναλήψεις και με ποιο βήμα προσαυξάνεται αυτή σε κάθε νέα επανάληψη. Στην συνέχεια να γραφεί ισοδύναμο πρόγραμμα με την χρήση της εντολής for αντί της while.

```
#include <iostream>
using namespace std;
int main ()
{
    int sum,oros,last;
    cin>>last;
    sum=0;
    oros=1;
    while (oros<=last)
    {
        sum=sum+oros;
        cout<<"the sum is "<<sum<<"\n";
        oros=oros+1;
    }
    cout <<" end of loop \n";
    return 0;
}
```

Έστω πως εισάγεται η τιμή 4 στην last. Τι θα εμφανίσει η εκτέλεση του προγράμματος;

(β) Ανοίξτε το πρόγραμμα **ConvertWhileAutocheck.cpp** και να μετατρέψετε την εντολή while σε ισοδύναμη εντολή for.

ΑΣΚΗΣΕΙΣ

- 1) Γράψτε ένα πρόγραμμα που θα εμφανίζει τους περιττούς αριθμούς από το 1 μέχρι το 99.
- 2) Γράψτε ένα πρόγραμμα που θα ζητά τα ύψη έξι φίλων και θα εμφανίζει το μεγαλύτερο ύψος. Θα χρειαστεί να χρησιμοποιήσετε την ίδια μεταβλητή (έστω height) για να εισαχθούν τα 6 ύψη.
- 3) Γράψτε ένα πρόγραμμα που θα ζητά το πλήθος των ατόμων σε μία παρέα. Θα διαβάζει τα ύψη κάθε ατόμου στην παρέα και θα εμφανίζει το μεγαλύτερο αλλά και το χαμηλότερο ύψος.
- 4) Γράψτε ένα πρόγραμμα που θα εμφανίζει τις πρώτες 10 δυνάμεις του 2. Η έξοδος θα πρέπει να είναι της μορφής $2^0 = 1$ $2^1 = 2$ $2^2 = 4$... $2^{10} = 1024$
- 5) Γράψτε ένα πρόγραμμα που θα διαβάζει 26 βαθμούς των μαθητών του τμήματός σας και θα καταμετρά και θα εμφανίζει πόσοι από αυτούς είναι άριστοι.
- 6) ΠΑΡΟΜΟΙΑ ΜΕ ΘΕΜΑ Α' ΦΑΣΗΣ ΠΔΠ. Έστω πως στέκονται 8 άτομα σε μία ουρά μπροστά σε ένα ταμείο θεάτρου. Αν υποθέσουμε πως το ταμείο το βλέπει μόνο ο πρώτος και όσα άτομα στην ουρά δεν έχουν άτομο στο ίδιο ή με μεγαλύτερο ύψος από τους ίδιους μπροστά τους. Να γράψετε πρόγραμμα που διαβάζει τα ύψη των ατόμων ξεκινώντας από τον πρώτο στην ουρά και εμφανίζει πόσοι από αυτούς βλέπουν το ταμείο. (πχ αν τα ύψη είναι 1.50 1.70 1.66 1.59 1.49 1.80 1.70 1.75, τρία άτομα θα βλέπουν το ταμείο)
- 7) Γράψτε ένα πρόγραμμα που θα διαβάζει 10 βαθμούς ενός μαθητή και θα εμφανίζει τον μέσο όρο της βαθμολογίας του. Να χρησιμοποιήσετε την ίδια μεταβλητή για την είσοδο των 10 βαθμών.
- 8) Γράψτε ένα πρόγραμμα που θα διαβάζει τους βαθμούς των μαθητών ενός τμήματός (άγνωστο πλήθος μαθητών) και θα εμφανίζει πόσοι από αυτούς είναι άριστοι. Καθώς δεν γνωρίζουμε πόσοι είναι οι βαθμοί, θα πρέπει να σταματά να διαβάζει βαθμό όταν εισαχθεί ως βαθμός αρνητικός αριθμός. (τιμή φρουρός θα είναι οποιαδήποτε αρνητική τιμή). Να υπολογίζει και να εμφανίζει το ποσοστό των άριστων βαθμολογιών.
- 9) Γράψτε ένα πρόγραμμα που θα διαβάζει θερμοκρασίες (άγνωστο πλήθος θερμοκρασιών) που κατεγράφησαν τον μήνα Δεκέμβριο και θα εμφανίζει τον μέσο όρο τους. Ως σημαία τερματισμού της εισαγωγής θερμοκρασιών θα είναι η πληκτρολόγηση της τιμής 100.

ΑΜΕΡΙΚΑΝΙΚΟ ΚΟΛΛΕΓΙΟ ΕΛΛΑΔΟΣ

- 10) Γράψτε ένα πρόγραμμα που θα διαβάζει τον κωδικό που απαιτείται για να ανοίξει ένα αρχείο ο οποίος είναι OpenSesame. Όσο ο κωδικός που εισάγεται είναι λάθος θα επαναλαμβάνεται η εντολή εισόδου ενώ όταν πληκτρολογηθεί σωστά θα εμφανίζεται το μήνυμα «You have opened this protected file in yourth attempt»
- 11) Γράψτε ένα πρόγραμμα που θα ζητά από τον χρήστη να εισάγει τον κωδικό πρόσβασής του (στην άσκησή μας τον *abc*). Δώστε στον χρήστη τη δυνατότητα να προσπαθήσει μέχρι 3 φορές να δώσει έγκυρο κωδικό. Μετά να του απαγορεύεται η είσοδος και να εμφανίζεται το μήνυμα *Access Denied – You have been locked out*.
- 12) Γράψτε ένα πρόγραμμα που θα εμφανίζει τις διαδοχικές δυνάμεις του 2 μέχρι την τιμή 8192. Επίσης να εμφανίζει τον εκθέτη k της τελευταίας δύναμης $2^k=8192$.
- 13) Γράψτε ένα πρόγραμμα που θα διαβάζει βαθμούς των μαθητών ενός τμήματός και θα εμφανίζει τον μέσο όρο της βαθμολογίας. Καθώς δεν γνωρίζουμε πόσοι είναι οι βαθμοί, θα πρέπει να σταματά να διαβάζει βαθμό όταν εισαχθεί ως βαθμός η τιμή φρουρός 999.
- 14) Γράψτε ένα πρόγραμμα που θα διαβάζει έναν βαθμό από το 0 έως το 20. Στην περίπτωση που διαβάσει βαθμό που βρίσκεται εκτός του διαστήματος θα πρέπει να διαβάζει επαναληπτικά νέο βαθμό μέχρι να εισαχθεί έγκυρος βαθμός. Όταν διαβάσει έγκυρο βαθμό να τον προσαυξήσει κατά μία μονάδα χωρίς ωστόσο να ξεπερνά το 20.
- 15) Το “99 bottles of beer” είναι ένα παραδοσιακό τραγούδι της βόρειας Αμερικής. Έχει ένα επαναλαμβανόμενο μοτίβο στους στίχους του και είναι εύκολο να το θυμούνται ακόμη και μικρής ηλικίας παιδιά. Χρειάζεται πολύ χρόνο για να ολοκληρωθεί και γι’ αυτό το τραγουδούν σε μεγάλα ταξίδια. Οι στίχοι του είναι:

*99 bottles of beer on the wall, 99 bottles of beer.
Take one down, pass it around, 98 bottles of beer on the wall.
98 bottles of beer on the wall, 98 bottles of beer.
Take one down, pass it around, 97 bottles of beer on the wall.
...
3 bottles of beer on the wall, 3 bottles of beer.
Take one down, pass it around, 2 bottles of beer on the wall.
2 bottles of beer on the wall, 2 bottles of beer.
Take one down, pass it around, 1 bottles of beer on the wall.
If that one bottle should happen to fall, what a waste of alcohol!*

Γράψτε ένα πρόγραμμα που θα εμφανίσει όλους τους στίχους στην οθόνη.

- 16) Γράψτε ένα πρόγραμμα που θα ζητά από τον χρήστη να επιλέξει μεταξύ των A, B, C και D. Το πρόγραμμα θα πρέπει να επαναλάβει τη διαδικασία μέχρις ότου ο χρήστης να πληκτρολογήσει μια έγκυρη επιλογή. Δείτε το παρακάτω στιγμιότυπο οθόνης για βοήθεια:

```
Select between A, B, C or D: F
Select AGAIN between A, B, C or D: d
Select AGAIN between A, B, C or D: D
Good choice!!!
Process returned 0 (0x0)   execution time : 17.075 s
Press any key to continue.
```

- 17) Γράψτε ένα πρόγραμμα που θα ζητά από τον χρήστη να εισάγει αριθμούς. Το πρόγραμμα θα τους προσθέτει και θα εμφανίζει το τελικό άθροισμά τους. Η διαδικασία εισαγωγής θα σταματάει όταν ο χρήστης εισάγει τον αριθμό μηδέν. (πχ η είσοδος 2 5 1 0 θα έχει ως έξοδο 8)

18) Να γράψετε πρόγραμμα που επιτρέπει την ανάληψη (withdrawal) ή την κατάθεση χρημάτων (deposit) από τραπεζικό λογαριασμό μέσω ATM. Με δεδομένο πως ο λογαριασμός τραπεζής έχει υπόλοιπο 480€ το πρόγραμμα να εμφανίζει αρχικά το μενού:

- | |
|---------------|
| 1. Withdrawal |
| 2. Deposit |
| 3. Exit |

και να ζητά την επιλογή του χρήστη 1 ή 2 ή 3.

Αν εισαχθεί 1 (ανάληψη), το πρόγραμμα θα ζητά το ποσό της ανάληψης και αν επαρκεί το υπόλοιπο θα εμφανίζει το νέο υπόλοιπο διαφορετικά θα εμφανίζει μήνυμα *Not enough money in account*. Στην συνέχεια θα ζητά από τον χρήστη να πληκτρολογήσει **Yes** αν επιθυμεί να κάνει New Transaction οπότε και θα επαναλαμβάνεται η εκτέλεση του αλγορίθμου.

Αν εισαχθεί 2 (κατάθεση), το πρόγραμμα θα ζητά το ποσό της κατάθεσης και θα εμφανίζει το νέο υπόλοιπο. Στην συνέχεια θα ζητά από τον χρήστη να πληκτρολογήσει **Yes** αν επιθυμεί να κάνει New Transaction οπότε και θα επαναλαμβάνεται η εκτέλεση του αλγορίθμου.

Αν εισαχθεί 3 (έξοδος), το πρόγραμμα θα εμφανίζει μήνυμα *Thank you for your preference to our Bank*.

19) Γράψτε ένα πρόγραμμα που καταμετρά τις απαντήσεις στην ερώτηση "What is your favourite color? Enter 1, 2, 3, 4 or 0 to exit poll." για μια έρευνα. Αρχικά θα εμφανίζει την ερώτηση που τίθεται μαζί με 4 επιλογές ως πιθανές απαντήσεις. Για παράδειγμα

1. Blue
2. Red
3. Yellow
4. Other

Θα καταμετρά τις απαντήσεις τους (1, 2, 3 ή 4) μέχρι να πληκτρολογήσει το 0, οπότε και θα σταματά η έρευνα. Τέλος θα εμφανίζει τα αποτελέσματα της καταμέτρησης καθώς και το ποσοστό των ερωτηθέντων που απάντησαν 4. Other.

20) Μια εταιρεία Πληροφορικής θέλει να υπολογίσει το μηνιαίο κόστος ανάπτυξης ενός Λογισμικού. Το κόστος αυτό υπολογίζεται βάσει του πίνακα:

Κατηγορία	Κόστος
προσωπικού	μήνα
Πληροφορικοί	2500€
Λοιπό προσωπικό	1800€

Να αναπτύξετε πρόγραμμα που για 40 άτομα που θα απασχοληθούν και για ένα μόνο μήνα:

- (α) διαβάσει επαναληπτικά τον κωδικό κατηγορίας του κάθε υπαλλήλου (διαβάσει είτε 0 για Πληροφορικούς, είτε 1 για Λοιπό προσωπικό)
- (β) εμφανίζει το πλήθος των Πληροφορικών που θα απασχοληθούν
- (γ) εμφανίζει το συνολικό κόστος του μήνα για την εταιρεία

Σκεφτείτε μία λύση που να εκτελεί όσο γίνεται λιγότερες εντολές έτσι ώστε ο χρόνος εκτέλεσης του προγράμματος να ελαχιστοποιείται.

- 21) Αναπτύξτε πρόγραμμα το οποίο θα ζητά από τον παίκτη να επιλέξει έναν μυστικό αριθμό από το 1 έως το 100. Το πρόγραμμα στην συνέχεια θα «μαντεύει» αριθμούς και ο παίκτης θα πρέπει να απαντά αν ο αριθμός που μάντεψε το πρόγραμμα είναι ίσος ή μεγαλύτερος από τον μυστικό αριθμό. Όσο το πρόγραμμα δεν έχει μαντέψει τον μυστικό αριθμό επαναλαμβάνει τις μαντεψιές. Όταν το πρόγραμμα βρει τον μυστικό αριθμό το παιχνίδι τερματίζει και εμφανίζει με πόσες μαντεψιές βρέθηκε.

Παράδειγμα

Πρόγραμμα: Επέλεξε έναν μυστικό αριθμό από το 1 έως το 100.

Πρόγραμμα: Είναι ο μυστικός αριθμός που επέλεξες 50;

Παίκτης: Όχι

Πρόγραμμα: Είναι ο μυστικός αριθμός μεγαλύτερος από 50;

Παίκτης: Όχι

Πρόγραμμα: Είναι ο μυστικός αριθμός που επέλεξες 25;

Παίκτης: Όχι

Πρόγραμμα: Είναι ο μυστικός αριθμός μεγαλύτερος από 25;

...

...

Πρόγραμμα: Βρήκα τον αριθμό με μαντεψιές.

Σκεφτείτε έναν αλγόριθμο με τον οποίο μπορεί ο υπολογιστής να ελαχιστοποιήσει πόσες μαντεψιές χρειάζεται να κάνει.

