

# 1. Εισαγωγή

## *Περιεχόμενα*

Η θεωρία περιλαμβάνει την κατανόηση των παρακάτω εννοιών:

- Machine Code (Γλώσσα Μηχανής)
- Assembly (Κωδικοποιημένη Γλώσσα Μηχανής)
- High level languages (Γλώσσες υψηλού επιπέδου)
- Source code (Πηγαίος κώδικας)
- Διερμηνευτής (Interpreter)
- Μεταγλωττιστής (compiler)
- Προεπεξεργαστής (Pre-Processor)
- Object code (Αντικειμενικός)
- Linker (Διασυνδέτης )
- Executable (Εκτελέσιμο πρόγραμμα)
- ANSI standard
- IDE (Integrated Development Environment)
- Το πρώτο μου πρόγραμμα (Hello.cpp)
- Λάθη μεταγλώττισης (compile time errors)
- Λάθη χρόνου εκτέλεσης (Compile time errors)
- Το πρόγραμμα HelloWorld
  - Ο Προεπεξεργαστής (Preprocessor)
  - Η οδηγία (προς τον προεπεξεργαστή) *include*
  - Το αρχείο *iostream* (Input Output stream)
  - Η συνάρτηση(function) *main*
  - Opening and enclosing brace
  - Τιμή επιστροφής (Return value) της *main*
  - Η standard library
  - Βασικό output με το (αντικείμενο) *cout*
  - Ο προσδιοριστής ονοματοχώρου (namespace specifier) *std*
  - Ο τελεστής ανακατεύθυνσης εξόδου (output redirection operator) <<
  - Σειρές (εκτυπώσιμων) χαρακτήρων (Strings)
  - Νέα γραμμή με τον χαρακτήρα διαμόρφωσης (formatting character) '\n'
- **Περισσότερη cout**

- Νέα γραμμή με την *endl* (end line)
- Αξιολόγηση των εκφράσεων (evaluation)
- Χαρακτήρες διαφυγής (escape sequences)
- **Η λέξη κλειδί (keyword) using (Listing 2.3)**
- **standard ονοματοχώρος (namespace) (Listing 2.4)**
- **Σχόλια (Comments)**
  - Σχόλια σε μια γραμμή
  - Σχόλια σε πολλαπλές γραμμές
- **Μεταβλητές-Τύποι**
  - Βασικοί τύποι (Native types)
  - Μεταβλητές (variables)
  - Σταθερές (constants)
  - Τελεστές (operators)
    - Πρόσθεση
    - Διαίρεση ακεραίων - πραγματικών
    - Μετατροπή τύπου – float, double
    - Τελεστές Προσαύξησης (Increment operators)
- **Εισαγωγή στις συναρτήσεις**
  - Συναρτήσεις χωρίς τιμή επιστροφής
  - Συναρτήσεις με παραμέτρους

## Σημειώσεις

Στην ενότητα αυτή παρουσιάζονται ορισμένες βασικές έννοιες που προαπαιτούνται στον προγραμματισμό. Επίσης, παρουσιάζονται τα πρώτα προγράμματα σε C++.

## Γλώσσα Μηχανής (Machine Code)

Η θεμελιώδης γλώσσα προγραμματισμού είναι η λεγόμενη Γλώσσα Μηχανής. Το “αλφάβητο” αυτής της γλώσσας είναι οι χαρακτήρες 0 και 1, δηλ. τα δυαδικά ψηφία (binary digits ή bits). Επομένως, τα προγράμματα που είναι γραμμένα σε γλώσσα μηχανής δεν είναι παρά ακολουθίες από bits. Η γλώσσα μηχανής είναι η μοναδική γλώσσα στην οποία ανταποκρίνεται η μηχανή, δηλ. το υλικό μέρος του υπολογιστή. Στην εικόνα 1 δίνεται ένα παράδειγμα γλώσσας μηχανής.

Assembly Language	Machine Code
add \$t1, \$t2, \$t3	04CB: 0000 0100 1100 1011
addi \$t2, \$t3, 60	16BC: 0001 0110 1011 1100
and \$t3, \$t1, \$t2	0299: 0000 0010 1001 1001
andi \$t3, \$t1, 5	22C5: 0010 0010 1100 0101
beq \$t1, \$t2, 4	3444: 0011 0100 0100 0100
bne \$t1, \$t2, 4	4444: 0100 0100 0100 0100
j 0x50	F032: 1111 0000 0011 0010
lw \$t1, 16(\$s1)	5A50: 0101 1010 0101 0000
nop	0005: 0000 0000 0000 0101
nor \$t3, \$t1, \$t2	029E: 0000 0010 1001 1110
or \$t3, \$t1, \$t2	029A: 0000 0010 1001 1010
ori \$t3, \$t1, 10	62CA: 0110 0010 1100 1010
ssl \$t2, \$t1, 2	0455: 0000 0100 0101 0101
srl \$t2, \$t1, 1	0457: 0000 0100 0101 0111
sw \$t1, 16(\$t0)	7050: 0111 0000 0101 0000
sub \$t2, \$t1, \$t0	0214: 0000 0010 0001 0100

Εικόνα 1: Γλώσσα Μηχανής

### *Κωδικοποιημένη Γλώσσα Μηχανής(Assembly)*

Η Assembly είναι η γλώσσα μηχανής κωδικοποιημένη. Πρόκειται επομένως για μια συμβολική χαμηλού επιπέδου γλώσσα προγραμματισμού, δηλαδή μια γλώσσα πολύ κοντά στη γλώσσα μηχανής και στο υλικό του υπολογιστή. Κάθε συγκεκριμένη αρχιτεκτονική συνόλου εντολών, δηλαδή κάθε οικογένεια επεξεργαστών, έχει τη δική της συμβολική γλώσσα, η οποία δίνεται συνήθως από τον κατασκευαστή της. Η Assembly είναι πιο ευανάγνωστη από την γλώσσα μηχανής και διευκολύνει τον άνθρωπο να κατανοεί την γλώσσα μηχανής.

Για παράδειγμα ένας επεξεργαστής της αρχιτεκτονικής [x86/IA-32](#) θα καταλάβει την εντολή σε γλώσσα μηχανής:

```
10110000 01100001
```

Ένας προγραμματιστής όμως είναι πιο εύκολο να θυμάται την ισοδύναμη συμβολική αναπαράσταση, για παράδειγμα μια τυπική εντολή σε συμβολική γλώσσα είναι η εξής:

```
mov al, 061h
```

που είναι συντομογραφία της αγγλικής λέξης move ("μετακίνησε"). Η εντολή αυτή μετακινεί τη δεκαεξαδική τιμή 61 (97 στο δεκαδικό σύστημα) στον καταχωρητή με το όνομα "al".

Η μετατροπή ενός προγράμματος από συμβολική γλώσσα σε γλώσσα μηχανής γίνεται από ένα συμβολομεταφραστή (assembler) και το αντίστροφο γίνεται από έναν αντισυμβολομεταφραστή (disassembler).

### ***Γλώσσες υψηλού επιπέδου (High level languages)***

Ως υψηλού επιπέδου γλώσσα προγραμματισμού (high-level programming language) ορίζεται αυτή που επιτρέπει τη μεταφερισιμότητα ενός προγράμματος από έναν υπολογιστή σε έναν άλλο. Αποτελείται από εντολές εύκολα κατανοητές στον προγραμματιστή, καθώς μοιάζουν με -περιορισμένη- φυσική γλώσσα. Για την εκτέλεση του προγράμματος από τον υπολογιστή, απαιτείται η χρήση μεταγλωττιστή για την παραγωγή του προγράμματος σε γλώσσα μηχανής. Παραδείγματα γλωσσών υψηλού επιπέδου : C++, Cobol, Pascal, κα.

### ***Πηγαίος κώδικας (Source code)***

Πηγαίος κώδικας είναι οποιαδήποτε σειρά από εντολές ή δηλώσεις σε κάποια γλώσσα υψηλού επιπέδου. Ο όρος πηγαίος κώδικας αναφέρεται συνήθως σε εντολές που γράφονται από κάποιον προγραμματιστή σε μια γλώσσα προγραμματισμού, και όχι σε εντολές που παράγονται αυτόματα από λογισμικό.

Ο πηγαίος κώδικας μπορεί να μεταγλωττιστεί σε εκτελέσιμο κώδικα μηχανής ή να εκτελεστεί ως έχει από κάποιον διερμηνευτή.

### ***Διερμηνευτής (Interpreter)***

Ο Διερμηνευτής δέχεται ως είσοδο έναν κώδικα υψηλού επιπέδου και μεταφράζει και εκτελεί μια-μια τις εντολές του κώδικα.

### ***Μεταγλωττιστής (compiler)***

Ο Μεταγλωττιστής δέχεται ως είσοδο έναν κώδικα υψηλού επιπέδου, τον μεταφράζει παράγοντας έναν κώδικα που ονομάζεται Object code (Αντικειμενικός).

Εισαγωγή στον Προγραμματισμό με την C++

### ***Linker (Διασυνδέτης)***

Ο διασυνδέτης διασυνδέει ένα σύνολο από αντικειμενικούς κώδικες και παράγει τον εκτελέσιμο (executable) κώδικα, δηλ. τον κώδικα μηχανής

### ***Προεπεξεργαστής (Pre-Processor)***

Ο προεπεξεργαστής κάνει κάποια επεξεργασία στον πηγαίο κώδικα, όπως για παράδειγμα αντικαθιστά το όνομα ενός αρχείου με το περιεχόμενό του, π.χ. οδηγία include. Όπως φανερώνει το όνομά του, η επεξεργασία αυτή εφαρμόζεται πριν την μεταγλώττιση.

### ***Απολαθοποιητής (Debugger)***

Ο debugger διευκολύνει στην εύρεση και διόρθωση λαθών του πηγαίου κώδικα. Ας σημειωθεί πως υπάρχουν 2 τύποι λαθών: Τα λάθη μεταγλώττισης (compile time errors) που εντοπίζει ο μεταγλωττιστής και επομένως είναι αντιμετωπίσιμα και τα λάθη χρόνου εκτέλεσης (run time errors) που δεν εντοπίζει ο μεταγλωττιστής και άρα εκδηλώνονται κατά τον χρόνο εκτέλεσης. Τα λάθη του χρόνου εκτέλεσης είναι πάρα πολύ σοβαρά καθώς περνάνε στον εκτελέσιμο κώδικα, φτάνουν στον χρήστη και είναι δυνατόν να προκαλέσουν σημαντικές καταστροφές, π.χ. λάθη χρόνου εκτέλεσης σε λογισμικό πλοήγησης αεροσκάφους.

### ***Πρότυπη C++***

Το Αμερικανικό Εθνικό Ίδρυμα Προτυποποίησης (American National Standards Institute, ANSI) ;έχει δημιουργήσει ένα πρότυπο για την C++ (ANSI standard). Έτσι διασφαλίζεται πως όλοι οι μεταγλωττιστές που ακολουθούν το πρότυπο μπορούν να μεταγλωττίσουν με επιτυχία έναν πηγαίο κώδικα C++ που επίσης ακολουθεί το πρότυπο. Ας σημειωθεί ότι ο ANSI είναι ο αμερικανικός εκπρόσωπος του Διεθνούς Οργανισμού Προτυποποίησης (International Organization for Standardization - ISO) και της Διεθνούς Ηλεκτροτεχνικής Επιτροπής (International Electrotechnical Commission - IEC). Προκειμένου να ακολουθήσουμε το ANSI πρότυπο σε αυτά τα μαθήματα, υιοθετούμε ως μεταγλωττιστή τον g++.

Εισαγωγή στον Προγραμματισμό με την C++

### ***Ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment, IDE)***

Παλιότερα, ο πηγαίος κώδικας γράφονταν με την βοήθεια ενός text editor. Στην συνέχεια, καλούνταν ο προεπεξεργαστής, παρήγαγε ένα νέο πηγαίο κώδικα. Ο κώδικας αυτός αποτελούσε είσοδο για τον μεταγλωττιστή ο οποίος παρήγαγε τον αντικειμενικό κώδικα. Τέλος, καλούνταν ο διασυνδέτης που ελάμβανε σαν είσοδο μια σειρά από αντικειμενικούς κώδικες προκειμένου να τους διασυνδέσει και να παράγει τον εκτελέσιμο κώδικα. Επίσης, ο απολαθοποιητής ήταν ένα ξεχωριστό λογισμικό που έπρεπε να κληθεί ανεξάρτητα όποτε υπήρχε η ανάγκη. Σήμερα τα πράγματα έχουν αλλάξει. Τα ολοκληρωμένα περιβάλλοντα ανάπτυξης προσφέρουν όλες αυτές τις λειτουργίες (και πολλές άλλες) ενσωματωμένες σε ένα περιβάλλον και διευκολύνουν με αυτόν τον τρόπο την ανάπτυξη των προγραμμάτων. Το IDE που θα χρησιμοποιήσουμε κατά την διάρκεια αυτών των μαθημάτων είναι το NetBeans.

### ***Το πρώτο μου πρόγραμμα (Hello.cpp)***

```
#include <iostream>

using namespace std;

int main(int argc, char** argv) {

    cout << "Hello World";

    return 0;

}
```

### ***Αλλαγή γραμμής***

```
#include <iostream>

using namespace std;
```

Εισαγωγή στον Προγραμματισμό με την C++

```
int main(int argc, char** argv) {  
    cout << "Hello World 1";  
    cout << "Hello World 2";  
    return 0;  
}  
  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char** argv) {  
    cout << "Hello World 1";  
    cout << endl;  
    cout << "Hello World 2";  
    return 0;  
}  
  
#include <iostream>  
  
using namespace std;  
  
int main(int argc, char** argv) {  
    cout << "Hello World 1" << endl;  
    cout << "Hello World 2";  
    return 0;  
}
```

### **Μεταβλητές**

Οι μεταβλητές είναι ένα κρίσιμο θέμα στην κατανόηση του προγραμματισμού. Έστω ο παρακάτω κώδικας:

1. `int i;`
2. `i=5;`
3. `cout << i;`

Στην γραμμή 1, δηλώνεται ότι θα χρησιμοποιηθεί μια μεταβλητή που θα δέχεται ακέραιες τιμές. Ο τύπος της μεταβλητής ορίζεται από την δεσμευμένη λέξη (reserved word) `int`, δηλ. `integer` (ακέραιος). Στην γραμμή 2, η `i` ορίζεται καθώς δεσμεύεται χώρος στη μνήμη ίσος με ένα ακέραιο και στον χώρο αυτόν εκχωρείται η τιμή 5. Στην γραμμή 3, υπολογίζεται η τιμή της μεταβλητής `i` και ωθείται στο ρεύμα εξόδου.

Ας σημειωθεί πως η δήλωση (γραμμή 1) και ο ορισμός μιας μεταβλητής (γραμμή 2) μπορεί να γίνουν σε μια εντολή όπως φαίνεται αμέσως παρακάτω.

```
int i=5;  
cout << i;
```

Και στις 2 περιπτώσεις ορίσαμε πως πρόκειται για ακέραιη μεταβλητή, ορίσαμε δηλ. τον τύπο της μεταβλητής. Πράγματι, η C++ είναι μια γλώσσα που κάνει αυστηρό έλεγχο στον τύπο των μεταβλητών (strong typing language). Εκτός από τον ακέραιο υποστηρίζεται και ο πραγματικός (`double`). Πρόκειται για ένα τύπο κινητής υποδιαστολής.

```
double d=3.8, d1=2.0;
```

Ο ακέραιος και ο τύπος κινητής υποδιαστολής είναι οι δύο θεμελιώδεις τύποι του συστήματος με την έννοια ότι κάθε άλλος βασικός τύπος είναι είτε ακέραιος είτε κινητής υποδιαστολής.



## Εισαγωγή στον Προγραμματισμό με την C++

Πως ακριβώς αναπαρίστανται στην μνήμη οι ακέραιοι και οι πραγματικοί είναι κάτι που θα συζητήσουμε αργότερα. Ωστόσο, είναι χρήσιμο να γνωρίζουμε πως το ακριβές μέγεθος που καταλαμβάνουν στην μνήμη δεν είναι γνωστό εκ των προτέρων καθώς εξαρτάται από τον μεταγλωττιστή και το υλικό. Έτσι η πληροφορία αυτή είναι διαθέσιμη κατά τον χρόνο εκτέλεσης με την βοήθεια της συνάρτησης `sizeof` που διαθέτει η C++. Για παράδειγμα, η κλήση

```
cout << sizeof(int);
```

θα τυπώσει στην οθόνη τον αριθμό των bytes που καταλαμβάνει στην μνήμη ο ακέραιος.

Άλλοι βασικοί τύποι που υποστηρίζονται είναι οι `char`, `bool` και `float`.

Ο `char` χρησιμοποιείται για την διαχείριση χαρακτήρων. Ο `char` είναι κατ' ουσία ακέραιος τύπος.

```
char c='A';
```

```
cout << c;
```

Για παράδειγμα, στον παραπάνω κώδικα, δηλώνεται μια μεταβλητή `c` τύπου `char`, δεσμεύονται `sizeof(char)` bytes και τοποθετείται ο ASCII κωδικός του χαρακτήρα `A` που είναι ακέραιος. Όταν καλείται η `cout`, τότε υπολογίζεται η τιμή του `c` η οποία ωθείται στο ρεύμα εξόδου.

Ο `bool` ως τύπος δέχεται 2 τιμές: `true` και `false`. Πρόκειται επίσης για ακέραιο τύπο. Οποιαδήποτε τιμή διαφορετική του 0 θεωρείται `true` ενώ το 0 θεωρείται `false`.

Ο τύπος `float` είναι ένας τύπος κινητής υποδιαστολής με χαμηλότερη ακρίβεια από τον `double`.

Εκτός από αυτούς τους βασικούς τύπους έχουμε και τους ακόλουθους μετατροπείς, `short`, `long`, `unsigned`. Έτσι, ο `short int` καταλαμβάνει λιγότερα bytes από τον απλό `int`, ο `long int` περισσότερα ή ίσα ενώ ο `unsigned int` αναφέρεται σε ακέραιους μη αρνητικούς.

Εισαγωγή στον Προγραμματισμό με την C++

### **Σταθερές**

Μερικές φορές σε ένα πρόγραμμα χρειαζόμαστε κάποιες σταθερές, π.χ.

```
FPA=0.13*axia;
```

Στην παραπάνω γραμμή η *axia* πολλαπλασιάζεται με την σταθερά 0.13 για να υπολογιστεί η τιμή του ΦΠΑ. Μια τέτοια σταθερά θα μπορούσαμε να δηλώσουμε με το keyword `const` και να χρησιμοποιούμε στην συνέχεια.

```
const double SYNT_FPA=0.13;
```

Το keyword `const` μας διασφαλίζει ότι δεν πρόκειται να αλλάξει η τιμή του `SYNT_FPA`. Με αυτόν τον τρόπο τα προγράμματά μας γίνονται πιο ευανάγνωστα και πιο συντηρήσιμα.

### **Τελεστές**

Στην C++ υποστηρίζονται οι εξής βασικοί τελεστές:

**Εκχώρηση:** Η εκχώρηση γίνεται με χρήση του τελεστή `=`, π.χ. `i=j`, όπου και οι 2 μεταβλητές είναι ίδιου τύπου. Εκχώρηση σε σταθερά ελέγχεται ως λανθασμένη από τον μεταγλωττιστή. Η εκχώρηση τοποθετεί στην μεταβλητή που βρίσκεται στα αριστερά του τελεστή `=` (lValue) το περιεχόμενο της μεταβλητής που βρίσκεται στα δεξιά (rValue). Επιπλέον, επιστρέφει την μεταβλητή στα αριστερά στο περιβάλλον στο οποίο κλήθηκε. Έτσι, υποστηρίζεται η πολλαπλή εκχώρηση, π.χ. `i=j=2`;

**Πρόσθεση:** Η Πρόσθεση είναι προφανής, π.χ. `synolo=poso1+poso2`;

**Αφαίρεση:** Επίσης, προφανής είναι η αφαίρεση και ο πολλαπλασιασμός.

**Διαίρεση:** Η διαίρεση διαφοροποιείται σε διαίρεση πραγματικών και διαίρεση ακεραίων. Για παράδειγμα,

1. `int i=2, j=3;`
2. `double d=2.0, b=3.0;`

## Εισαγωγή στον Προγραμματισμό με την C++

3. `cout << j/l << endl;`
4. `cout << b/d << endl;`

το αποτέλεσμα στην γραμμή 3 είναι 1 ενώ στην γραμμή 4 είναι 1.5. Επίσης, προσφέρεται και ο τελεστής % που μας δίνει το υπόλοιπο της ακεραίας διαίρεσης.

Προσαύξηση: Ας θεωρήσουμε τον ακόλουθο κώδικα `k=k+2`; Πρόκειται για 2 πράξεις σε μια εντολή, μια πρόσθεση και μια εκχώρηση. Η C++ μας δίνει μια συντομογραφία για αυτήν την λειτουργία με τον τελεστή `+=`. Πιο συγκεκριμένα, `k=k+2` είναι ισοδύναμη με `k+=2`; Παρομοίως ισχύει η μείωση, π.χ. `k-=3`, ο πολλαπλασιασμός με εκχώρηση, `k*=3` και η διαίρεση με εκχώρηση, `k/=2`;

### **Χαρακτήρες διαφυγής**

Σε πολλές περιπτώσεις σε μια αλφαριθμητική σταθερά υπάρχει η ανάγκη να τοποθετηθεί ένας ή περισσότεροι χαρακτήρες διαφυγής (escape sequences). Για παράδειγμα

```
cout << "Hello Mary \n";  
cout << "How are you?";
```

Με το `\` σηματοδοτείται πως ο χαρακτήρας που ακολουθεί δεν έχει την συνηθισμένη σημασία του. Στην προκειμένη περίπτωση, η ειδική σημασία του χαρακτήρα `n` είναι η σηματοδότηση τέλους γραμμής της εξόδου. Άλλοι χαρακτήρες διαφυγής είναι:

`\t` : αλλαγή στήλης

`\r`: carriage return

### **Ασκήσεις με Λύσεις**

1. Να αναπτυχθεί ένα πρόγραμμα C++ που δέχεται ένα πραγματικό αριθμό που αντιπροσωπεύει ένα ποσό (Κεφάλαιο) και στην συνέχεια το πολλαπλασιάζει επί μια σταθερά που αντιπροσωπεύει το επιτόκιο ώστε να υπολογίσει και να τυπώσει τον τόκο
2. Να αναπτυχθεί ένα πρόγραμμα C++ που δέχεται δυο ακέραιες τιμές, τον διαιρετέο και τον διαιρέτη και στην συνέχεια υπολογίζει και τυπώνει το πηλίκο και το υπόλοιπο της ακεραίας διαίρεσης διαιρετέου δια διαιρέτη.

3. Να εκτυπωθούν τα ονοματεπώνυμα

Lefteris	Moussiades
Maria	Kouri
Iwannhs	Maragkos
Nikolaos	Apostolidhs

σε στήλες όπως φαίνονται εδώ.

4. Να αναπτυχθεί πρόγραμμα που λαμβάνει είσοδο από τον χρήστη έναν ακέραιο  $n$  και τυπώνει στην οθόνη τους από 1 έως  $n$ .
5. Να αναπτυχθεί πρόγραμμα που λαμβάνει είσοδο από τον χρήστη έναν ακέραιο  $n$  και τυπώνει στην οθόνη τους από 1 έως  $n$  που διαιρούνται ακριβώς με το 3 και το 7.
6. Να αναπτυχθεί πρόγραμμα που λαμβάνει είσοδο από τον χρήστη  $n$  ακεραίους και τυπώνει στην οθόνη τον μεγαλύτερο και τον μικρότερο.
7. Να αναπτυχθεί πρόγραμμα που τυπώνει στην οθόνη τις ονομασίες των καρτών μιας τράπουλας.
8. Να αναπτυχθεί πρόγραμμα που υπολογίζει το  $n!$  (παραγοντικό)
9. Γράψτε πρόγραμμα που τυπώνει ένα βέλος με φορά προς τα πάνω όπως φαίνεται παρακάτω

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

Λύσεις

```
#include <iostream>  
using namespace std;  
  
void x1() {  
    const double RATE = 0.06;  
    double capital;  
    cout << "Dwse kefalaiο : ";
```

## Εισαγωγή στον Προγραμματισμό με την C++

```
cin >> capital;
cout << "O tokos einai : " << capital*RATE;
}
```

```
void x2() {
    int diaireteos, diaireths;
    cout << "Dwse diaireteo : ";
    cin >> diaireteos;
    cout << "Dwse diaireh : ";
    cin >> diaireths;
    cout << "H akeraih diaresh tou " << diaireteos << " dia tou "
        << diaireths << "\ndinei phliko " << diaireteos / diaireths
        << " kai ypoloipo " << diaireteos % diaireths;
}
```

```
void x3() {
    cout << "Lefteris\tMoussiades\n"
        << "Maria\t\tKouri\n"
        << "Iwannhs\t\tMaragkos\n"
        << "Nikolaos\tApostolidhs";
}
```

```
void x4() {
    int n;
    cout << "Dwse n : ";
    cin >> n;
    cout << endl;
    for (int i = 1; i <= n; i++) cout << i << " ";
    cout << endl;
}
```

```
void x5() {
    int n;
    cout << "Dwse n : ";
    cin >> n;
    cout << endl;
    for (int i = 1; i <= n; i++)
        if (!(i % 3 == 0) && (i % 7 == 0)) cout << i << " ";
    cout << endl;
}
```

```
void x6() {
    int plithos;
    cout << "Dwse plithos : ";
    cin >> plithos;
    cout << endl;
}
```

```
int max, min, aa = 0;
int cNum;
for (int i = 0; i < plithos; i++) {
    cin >> cNum;
    if (aa == 0) {
        max = cNum;
        min = cNum;
    } else {
        if (cNum > max) max = cNum;
        if (cNum < min) min = cNum;
    }
    aa++;
}
cout << "max = " << max << "\nmin = " << min;
}
```

```
void x7() {
    for (int i = 1; i <= 13; i++)

        for (int j = 0; j < 4; j++) {
            switch (i) {
                case 1: cout << "Ace";
                    break;
                case 2: cout << "Two";
                    break;
                case 3: cout << "Three";
                    break;
                case 4: cout << "Four";
                    break;
                case 5: cout << "Five";
                    break;
                case 6: cout << "Six";
                    break;
                case 7: cout << "Seven";
                    break;
                case 8: cout << "Eight";
                    break;
                case 9: cout << "Nine";
                    break;
                case 10: cout << "Ten";
                    break;
                case 11: cout << "Vales";
                    break;
                case 12: cout << "Dama";
                    break;
                case 13: cout << "Rigas";
                    break;
            }
        }
}
```

```
    }  
    switch (j) {  
        case 0: cout << " koupa";  
            break;  
        case 1: cout << " karo";  
            break;  
        case 2: cout << " bastouni";  
            break;  
        case 3: cout << " spathi";  
            break;  
    }  
    cout << endl;  
}  
}
```

```
void factorial() {  
    long int paragontiko=1;  
    int n;  
    cout << "Dwse arithmo : ";  
    cin >> n;  
    cout << endl;  
    if (n>1)  
        for (int i=2; i<=n; i++) paragontiko*=i;  
    cout << paragontiko << endl;  
}
```

```
void prt01(int k) {  
    for (int i=1; i<=k; i++) {  
        for (int j=0; j<i; j++)  
            cout << "*";  
        cout << endl;  
    }  
    for (int i=k; i>=1; i--) {  
        for (int j=0; j<i; j++)  
            cout << "*";  
        cout << endl;  
    }  
}
```

```
void prtVelos() {  
    cout << " * " << endl;  
    cout << " *** " << endl;  
    cout << " ***** " << endl;  
    cout << " ***** " << endl;  
    cout << "*****" << endl;  
    cout << " *** " << endl;
```

Εισαγωγή στον Προγραμματισμό με την C++

```
cout << " *** " << endl;
cout << " *** " << endl;
}
```

```
int main(int argc, char** argv) {
    // x1();
    //x2();
    //x3();
    x4();
    return 0;
}
```

### Ασκήσεις

10. Μελετήστε τον παρακάτω κώδικα

```
void x() {
    int i = 5;
    double d1 = 112.5, d2 = 3;
    i = d1;
    cout << i << " " << d1 << endl;
    char c = 'A';
    d1 = c;
    cout << c << " " << d1;
}
```

11. Μελετήστε τον παρακάτω κώδικα

```
void prt01(int k) {
    for (int i=1; i<=k; i++) {
        for (int j=0; j<i; j++)
            cout << "*";
        cout << endl;
    }
    for (int i=k; i>=1; i--) {
        for (int j=0; j<i; j++)
            cout << "*";
        cout << endl;
    }
}
```

12. Γράψτε πρόγραμμα που τυπώνει το ονοματεπώνυμό σας.

13. Γράψτε πρόγραμμα που τυπώνει το ονοματεπώνυμό σας και τα στοιχεία της διεύθυνσής σας.



## Εισαγωγή στον Προγραμματισμό με την C++

14. Γράψτε πρόγραμμα που τυπώνει τους αριθμούς από το 1 έως το 5, (α) οριζόντια και (β) κάθετα.
15. Γράψτε ένα πρόγραμμα που τυπώνει ένα βελάκι με φορά προς τα κάτω

## 2. Επιλογή και Επαναληπτικές Διαδικασίες

### *Περιεχόμενα*

- Εντολές (commands, statements)
- Εκφράσεις
- Τελεστές
- Μαθηματικοί
- Εκχώρησης
- Προσαύξησης και μείωσης
- Προτεραιότητα αξιολόγησης
- Ο τύπος bool
- If statement
- Λογικοί τελεστές
- Βραχεία αξιολόγηση (short circuit evaluation)
- Ο τριαδικός συντελεστής συνθήκης (ternary conditional operator)
- Τι είναι η συνάρτηση (function)
- Τιμή επιστροφής
- Λίστα παραμέτρων
- Τυπικές (formal) και πραγματικές (actual) παράμετροι (arguments)
- Function prototype
- Τοπικές μεταβλητές (local variables)
- Καθολικές μεταβλητές (global variables)
- Default parameters
- Υπερφόρτωση συναρτήσεων (Overloading Functions)
- Αναδρομή (Recursion)
- do -while
- while
- for
- break
- continue
- switch

## Κώδικες προς μελέτη

### Κώδικας 1

```
#include <iostream>
using namespace std ;

double get () {
    double num ;
    cout << "Enter num to sum or 0 to escape " ;
    cin >> num ;
    cout << endl ;
    return num ;
}

void sum () {
    double num , s = 0 ;
    do {
        num = get () ;
        if (num < 0 ) continue ;
        s += num ;
    }
    while (num != 0 ) ;
    cout << "-----" << endl ;
    cout << " " << s << endl ;
}

void prtInts (int from, int to) {
    while (from < to) cout << from++ << " " ;
}

void prtInts2 (int from, int to) {
    while (true) {
        if (from >= to) break ;
        cout << from++ << " " ;
    }
}

void forTst(int upVal) {
    for (int i = 0 ; i < upVal ; i++) cout << i << " " ;
    cout << endl ;
    for (int i = upVal-1 ; i >= 0 ; i--) cout << i << " " ;
    cout << endl ;
    for (char c = 'a' ; c <= 'z' ; c++) cout << c << " " ;
    cout << endl ;
    for (int i = 0 ; i < upVal ; i += 2) cout << i << " " ;
    cout << endl ;
    for (int i = 0 ; true ; i++) {
```

```
        if (i>=upVal) break;
        cout << i << " ";
    }
    cout << endl ;
}

void chkParameter(int p) {
    if (p<=0 || p>3)
        cout << "parameter is out of limits" << endl ;
    else
        if (p==1) cout << "Job 1" << endl ;
        else
            if (p==2) cout << "Job 2" << endl ;
            else
                if (p==3) cout << "Job 3" << endl ;
}

void chkParam(int p) {
    cout << p << " ";
    switch (p) {
        case (1) : cout << "Job 1" << endl ; break;
        case (2) : cout << "Job 2" << endl ; break;
        case (3) : cout << "Job 3" << endl ; break;
        default : cout << "parameter is out of limits" << endl ;
    }
}

int main()
{
    //sum();
    //prtInts2(0,10);
    //forTst(10);
    //for (int i=0; i<5; i++) chkParameter(i);
    for (int i=0; i<5; i++) chkParam(i);
    return 0;
}
```

## Κώδικας 2

```
#include <iostream>
#include "myFile.h"
using namespace std;

void checkIf() {
    cout << "checkif is running" << endl ;
    const int male= 1;
    const int female= 2;
```

```
//const int child=3;
int filo= male;
if ( filo==male) {
    cout << "He is man" << endl ;
    cout << "He is my friend" << endl ;
}
else if ( filo==female)
    cout << "She is woman" << endl ;
else cout << "it's a child" << endl ;
    cout << "checkIf stops" << endl ;
}

void checkIf2() {
    cout << "checkif2 is running" << endl ;
    const int male= 1;
    const int female= 2;
    const bool isMyFriend= false;
    int filo= female; bool filoS= isMyFriend;
    if ( filo==male)
        if ( filoS) cout << "he is my friend" << endl ;
        else cout << "he is not my friend" << endl ;
    else
        if ( filoS) cout << "She is my friend" << endl ;
        else cout << "She is not my friend" << endl ;
    cout << "checkIf2 stops" << endl ;
}

void checkFor() {
    for ( int i = 0 ; i < 10; i ++ ) cout << i << endl ;
    for ( int i = 4; i > 10; i = i +1 ) cout << i << endl ;
    for ( char c= 'a'; true; c++) {
        cout << c;
        if ( c>='z') break;
        if ( c=='k') { cout << endl ; continue;}
        cout << c << endl ;
    }
    for ( char c= 'A'; c<='Z'; c++)
        cout << c << " " << ( int )c << endl ;
}

void checkWhile() {
    int i = 0 ;
    while ( true) {
        if ( i ==2 || i ==4) { i ++; continue;}
        cout << i ;
        i ++;
        if ( i >9) break;
    }
}
```

```
    }
    cout << endl ;
    i = 0 ;
    while ( i < 10) cout << i ;
    cout << endl ;
    i = 0 ;
    while (++i < 10) cout << i ;
}

void checkIncrement() {
    int i=0;
    cout << ++i << " ";
    cout << i;
}

void checkDo() {
    char c='a';
    do {
        cout << ++c;
        //c++;
    }
    while (c<='z');
}

void localOrGlobalScope() {
    for (int i=0; i<5; i++) cout << i;
    cout << endl;
    int j;
    for (j=0; j<5; j++) cout << j;
    cout << endl;
    cout << j;
}

void compareBasicLoops() {
    for (int i=0; i<5; i++) cout << i;
    cout << endl;
    int i=0;
    while (i<5) cout << i++;
    cout << endl;
    i=0;
    do cout << i++;
    while (i<5);
}

int main() {
    //checkIf();
    //checkIf2();
}
```

```
//checkFor();  
//checkWhile();  
//checkDo();  
//checkIncrement();  
compareBasicLoops();  
return 0;  
}
```

### Ασκήσεις

16. Να αναπτυχθεί συνάρτηση που λύνει την γραμμική εξίσωση,  $a*x+b=0$ .
17. Να αναπτυχθεί συνάρτηση που λαμβάνει την ακέραιη παράμετρο  $k$  και επιστρέφει `true` αν η  $k$  είναι άρτιος, διαφορετικά `false`.
18. Να αναπτυχθεί συνάρτηση `double max(double a, double b)`; που λαμβάνει 2 πραγματικές παραμέτρους και επιστρέφει την τιμή της μεγαλύτερης. Στην `main` να εισαχθούν από το πληκτρολόγιο 3 πραγματικές τιμές και να βρεθεί η μεγαλύτερη με αξιοποίηση της `max`.
19. Να αναπτυχθεί συνάρτηση που λαμβάνει την ακέραιη παράμετρο  $k$  και τυπώνει τους ακεραίους  $0, 1, 2, \dots, k$  αν  $k$  μη αρνητική, διαφορετικά τυπώνει  $-1, -2, \dots, k$ .
20. Να αναπτυχθεί συνάρτηση `void prt01(int k)` που τυπώνει ένα τρίγωνο από αστεράκια όπως δείχνει το παρακάτω σχήμα:

prt01(5)

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

21. Να αναπτυχθεί συνάρτηση που επιστρέφει το παραγοντικό της ακέραιης παραμέτρου της.

22. Να αναπτυχθεί συνάρτηση που λαμβάνει την ακέραιη παράμετρο  $k$  και τυπώνει την αγγλική ονομασία του τελευταίου ψηφίου της  $k$ .