

Π. Σταθοπούλου

pstath@ece.upatras.gr ή
pstath@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

- Διδασκαλία : Παρασκευή 11πμ-13μμ ΗΛ7
- Φροντιστήριο : Δευτέρα 11πμ-12πμ ΗΛ4

Προηγούμενη Διάλεξη

- Εισαγωγικά
 - Διαδικαστικός προγραμματισμός, Διεργασία,
 - Οργάνωση της Λύσης, Διάγραμμα ροής,
 - Αλγόριθμος, Αναπαράσταση Δεδομένων,
- Δομή προγράμματος σε C
 - Συνάρτηση main, Δηλώσεις μεταβλητών,
 - Εκτελέσιμες προτάσεις.
- Εργαστήριο 1ο

1^ο Εργαστήριο

- Εξοικείωση με το προγραμματιστικό περιβάλλον
 - Τρόπος ανάπτυξης εφαρμογής,
 - Τρόπος παράδοσης ασκήσεων του εργαστηρίου
 - Προετοιμασία του 2^{ου} εργαστηρίου
- Δεν αξιολογείται
- Συνίσταται σε δύο τμήματα:
 - Υλοποίηση και παράδοση απλών προγραμμάτων
 - Εκτέλεση του Προγράμματος RobotLabyrinth

1^ο Εργαστήριο

- Τμήμα 1^ο - Σύνταξη, μεταγλώττιση και εκτέλεση
 - των προγραμμάτων των ενοτήτων 1.2.2 ή 1.6 (σελίδα 27 & 44)
 - του προγράμματος της ενότητας 3.4 (σελίδα 68)
 - του προγράμματος του παραδείγματος 3-3 (σελίδα 94) &
 - του προγράμματος της άσκησης 1 του κεφαλαίου 11 (σελίδα 326)

1^ο Εργαστήριο

Πρόγραμμα της ενότητας 1.2.2(σελίδα 27)

```
/* A simple C program */  
#include <stdio.h>  
  
main()  
{  
printf("hello, world");  
}
```

Σύνταξη,
μεταγλώττιση &
εκτέλεση

Περιβάλλον του DEV
C/C++

Περιβάλλον του DEV C/C++

Bloodshed Dev-C++ 4.9.9.2

Ο Bloodshed Dev-C++ 4.9.9.2 είναι ολοκληρωμένο περιβάλλον ανάπτυξης για

- σύνταξη
- μεταγλώττιση και
- εκτέλεση προγραμμάτων σε C ή C++

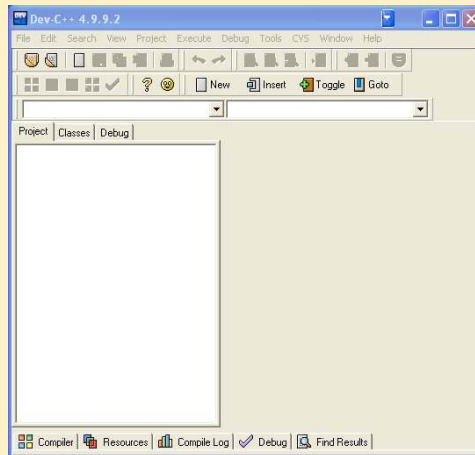
✓ τρέχει Windows 95

• 8 megabytes RAM, 30 megabytes σκληρό δίσκο

Μεγαλύτερη απόδοση

- ✓ 32 megabytes RAM, 45 megabytes σκληρό δίσκο &
- ✓ 233 MHz

Περιβάλλον του DEV C/C++



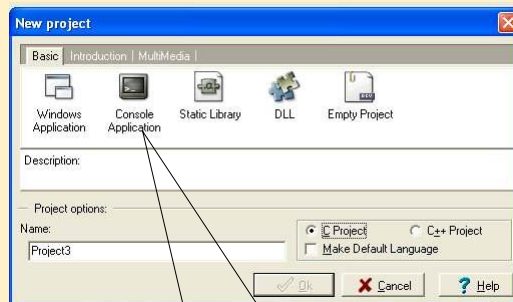
Περιβάλλον του DEV C/C++

```
#include <stdio.h>
#include <stdlib.h>

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

printf("hello, world");

system("PAUSE");
return 0;
}
```



- Μετά την επιλογή C project
- Επιλογή console Application

Πρόγραμμα της ενότητας 1.6 (σελίδα 44)

```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int a, b, c, sum;
    a=1;
    b=2; c=3;
    sum = a + b + c;
    printf("sum is %d",sum);

    system("PAUSE");
    return 0;
}
```

μεταγλώττιση



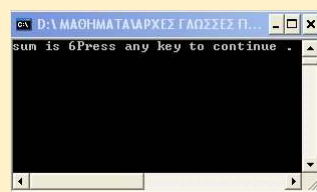
Πρόγραμμα της ενότητας 1.6 (σελίδα 44)

```
#include <stdio.h>
#include <stdlib.h>

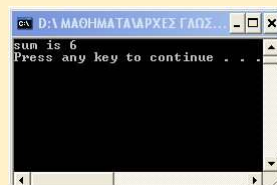
int main(int argc, char *argv[])
{
    int a, b, c, sum;
    a=1;
    b=2; c=3;
    sum = a + b + c;
    printf("sum is %d",sum);

    system("PAUSE");
    return 0;
}
```

εκτέλεση



printf("sum is %d\n",sum);



Πρόγραμμα του παραδείγματος 3.4 (σελίδα 68)

1. Ζήτα από τον χρήστη ένα χαρακτήρα
2. Πάρε από τον χρήστη τον χαρακτήρα
3. Τύπωσε τον χαρακτήρα και τον ASCII κωδικό του
4. Βρες τον επόμενο χαρακτήρα
5. Τύπωσε τον μαζί με τον κωδικό του

- ✓ scanf() → εντολή εισόδου από το πληκτρολόγιο
- ✓ scanf("format string", var₁, var₂,...var_n)
- ✓ format string →

%c για απλό χαρακτήρα
%d για ακέραιο
%f για κινητής υποδιαστολής
%s για συμβολοσειρά

Πρόγραμμα της ενότητας 3-3 (σελίδα 94)

```
#include <stdio.h>
#include <stdlib.h>

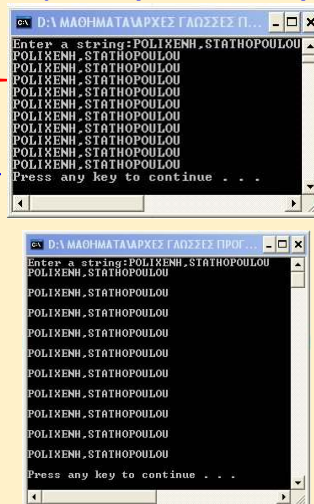
#define MAX_CHAR 80

int main(int argc, char *argv[])
{
    char str[MAX_CHAR];
    int i;

    printf("Enter a string:");
    scanf("%s",str);

    for (i=0;i<10;i++)
        printf("%s\n",str);

    system("PAUSE");
    return 0;
}
```



Εντολές Προεπεξεργαστή

• #include

Αναγκάζει τον μεταγλωττιστή (compiler) να συμπεριλάβει κάποια αρχεία
συνήθως πρόκειται για αρχεία της τυπικής βιβλιοθήκης της C

Τα αρχεία της τυπικής Βιβλιοθήκης έχουν ομαδοποιηθεί κατά κατηγορίες και δηλώνονται σαν διαφορετικά αρχεία κεφαλίδας.

```
<ctype.h>    <math.h>        <stdio.h>  
<stdlib.h>   <string.h>
```

Συντάσσεται και με αρχεία **#include "test.h"**
ΠΡΟΣΟΧΗ: δεν τερματίζει με ερωτηματικό(;))

Εντολές Προεπεξεργαστή

• #define

Ορίζει ένα αναγνωριστικό (Identifier) και ένα σύνολο χαρακτήρων που θα αντικαταστήσει αυτό το αναγνωριστικό, κατά την διαδικασία ον μεταγλώττισης του πηγαίου κώδικα.

Συντάσσεται ως εξής:

#define αναγνωριστικό χαρακτήρες
Π.χ. #define PI 3.141593

✓ Σημαντική η χρήση του για συγκεντρωτικές αλλαγές

✓ **ΠΡΟΣΟΧΗ:** δεν τερματίζει με ερωτηματικό(;))

Εντολές printf() & scanf()

Η συνάρτηση **printf()**

δίνει την δυνατότητα για τυποποιημένη εμφάνιση πληροφοριών στην οθόνη του Η/Υ.

Συντάσσεται ως εξής:

```
printf("format string", var1, var2, ..varn)
```

✓ format string →

%c	για απλό χαρακτήρα
%d	για ακέραιο
%f	για κινητής υποδιαστολής
%s	για συμβολοσειρά

✓ χαρακτήρες διαφυγής (escape characters) →

\n new line	\f form feed
\b backspace	\r carriage return
\t tab	\0 null character

Εντολή for – εντολή επανάληψης

Η εντολή **for** (ή βρόχος **for**)

χρησιμοποιείται για να περιγράψει επαναληπτικές διαδικασίες

Συντάσσεται ως εξής:

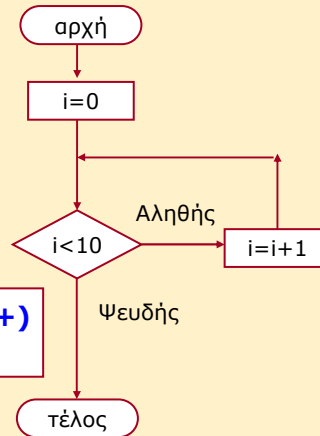
```
for (αρχική; συνθήκη; αύξηση)
```

- ▣ Η έκφραση *αρχική* είναι μία πρόταση εκχώρησης
- ▣ Η έκφραση *συνθήκη* είναι μία σχεσιακή έκφραση
 - ▣ Εάν η *συνθήκη* είναι ψευδής η πρόταση **for** τερματίζεται
 - ▣ Εάν η *συνθήκη* είναι αληθής εκτελούνται οι επόμενες προτάσεις
- ▣ Υπολογίζεται η έκφραση *αύξηση* και η εκτέλεση επιστρέφει στο 2^ο βήμα

Εντολή **for** – εντολή επανάληψης

```
for (i=0;i<10;i++)  
    printf("%s\n",str);
```

```
for (count=1;count<=20; count++)  
    printf("%d\n",count);
```



Πρόγραμμα της άσκησης 1 κεφάλαιο 11(σελίδα 326)

Έστω δύο αριθμοί x,y

1. Υπολογίζει το άθροισμα
2. Βρίσκει τον μεγαλύτερο
3. Βρίσκει το x^y
4. Υπολογίζει τους δεκαδικούς λογάριθμους των **x** και **y**

Τελεστές

- Τελεστής εκχώρησης
- Μαθηματικοί τελεστές
- Σχισιακοί τελεστές
- Λογικοί Τελεστές

```
#include <math.h>  
double pow(double x, double y)  
double log10(double x)
```

Τελεστές

Τελεστής Εκχώρησης

=
Π.χ. x=y;

Λογικοί Τελεστές

AND &&
OR ||
NOT !

Μαθηματικοί Τελεστές

- Μοναδιαίους
 - ++ τελεστής αύξησης,
 - τελεστής μείωσης,
- Διαδικούς
 - + πρόσθεση, - αφαίρεση,
 - / διαίρεση,
 - % υπόλοιπο διαίρεσης
 - * πολλαπλασιασμός

Σχεσιακοί Τελεστές

==
>
<
>=
<= \ !=

Εκτέλεση του Προγράμματος RobotLabyrinth

RobotLabyrinth.zip

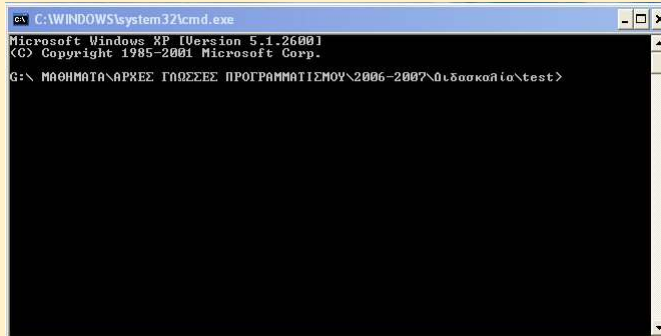
- RRController_amalia.zip
- RRController_cygwin.zip
- RRController_windows.zip

Αποσυμπιέζουμε το αρχείο *RobotLabyrinth.zip*



RobotLabyrinth.jar
labyrinth10.lab
labyrinth12.lab
labyrinth5.lab
labyrinth6.lab

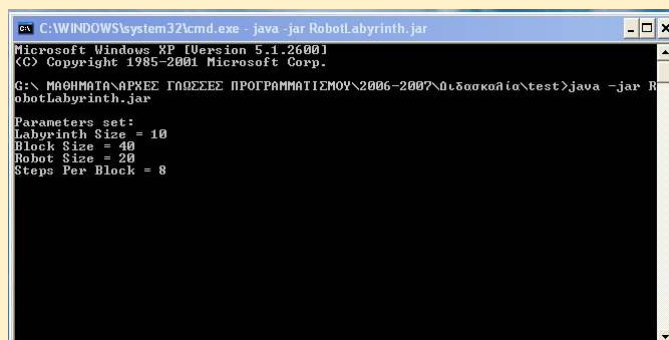
Εκτέλεση του Προγράμματος RobotLabyrinth



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
G:\ΜΑΘΗΜΑΤΑ\ΑΡΧΕΣ ΓΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ\2006-2007\διδασκαλία\test>
```

>java -jar RobotLabyrinth.jar

Εκτέλεση του Προγράμματος RobotLabyrinth

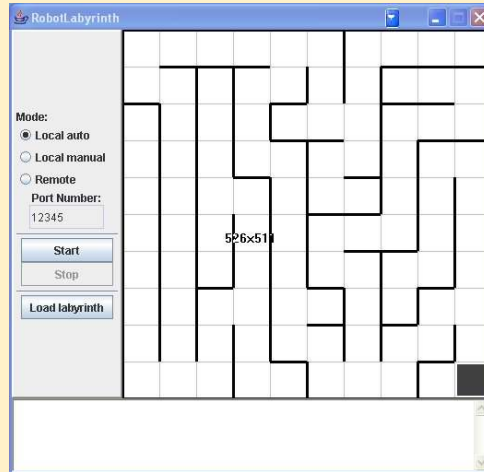


```
C:\WINDOWS\system32\cmd.exe - java -jar RobotLabyrinth.jar
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
G:\ΜΑΘΗΜΑΤΑ\ΑΡΧΕΣ ΓΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ\2006-2007\διδασκαλία\test>java -jar R
obotLabyrinth.jar
Parameters set:
Labyrinth Size = 10
Block Size = 40
Robot Size = 20
Steps Per Block = 8
```

Εκτέλεση του Προγράμματος RobotLabyrinth



- LOCAL AUTO
- LOCAL MANUAL
- REMOTE



Εκτέλεση του Προγράμματος RobotLabyrinth

- REMOTE RRController IP address & Port

