

Π. Σταθοπούλου

psth@ece.upatras.gr ή
psth@upatras.gr

Ομάδα Α' (Φοιτητές με μονό αριθμό Μητρώου)

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

Προηγούμενη Διάλεξη

Χρήση και Κατανόηση των Δεικτών

- * **Τι είναι δείκτης ?**
- * **Δημιουργία δείκτη**
- * **Δείκτες & απλές μεταβλητές**
- * **Απόδοση αρχικών τιμών σε δείκτες**
- * **Δείκτες & τύποι μεταβλητών**

Σήμερα

Ανάγνωση & εγγραφή Πληροφοριών

Διαχείριση των Ροών για Είσοδο/Έξοδο

Εμβέλεια Μεταβλητών

Ανάγνωση & Εγγραφή Πληροφοριών

Ανάγνωση
Πληροφοριών

Είσοδος Πληροφοριών από
τον εξωτερικό κόσμο στον
Υπολογιστή

Εγγραφή
Πληροφοριών

Έξοδος Πληροφοριών από
τον Υπολογιστή στον
εξωτερικό κόσμο

Ανάγνωση & Εγγραφή Πληροφοριών

Είσοδος Πληροφοριών από
τον εξωτερικό κόσμο στον
Υπολογιστή

Κυρίως με το
Πληκτρολόγιο

Έξοδος Πληροφοριών από
τον Υπολογιστή στον
εξωτερικό κόσμο

Κυρίως με την
Οθόνη

Ανάγνωση & Εγγραφή Πληροφοριών

Είσοδος Αριθμητικών δεδομένων

- Με την συνάρτηση της τυπικής βιβλιοθήκης της C
scanf()
- * διαβάζονται τα δεδομένα από το πληκτρολόγιο και εκχωρούνται σε μία ή περισσότερες μεταβλητές
- * Χρησιμοποιείται μια ακολουθία μορφής χαρακτήρων για να καθορισθεί η μορφή εισόδου.

Ανάγνωση & Εγγραφή Πληροφοριών

Η συνάρτηση **scanf()** όταν χρησιμοποιείται θα πρέπει να συμπεριλάβει και το αρχείο επικεφαλίδας **stdio.h**

Συντάσσεται ως εξής:

**scanf(ακολουθία-χαρακτήρων-μορφής
[ορίσματα,...])**

Τα ορίσματα είναι οι διευθύνσεις των μεταβλητών και όχι οι ίδιες οι μεταβλητές

Ανάγνωση & Εγγραφή Πληροφοριών

Προβολή πληροφοριών στην Οθόνη

➤ Με την συνάρτηση της τυπικής βιβλιοθήκης της C

printf(), puts()

- * Προβάλλονται μηνύματα κειμένου και τιμές μεταβλητών που εξάγουν τα προγράμματα.
- * Χρησιμοποιείται μια ακολουθία μορφής χαρακτήρων για να καθορισθεί η μορφή εμφάνισης της εξόδου.

Ανάγνωση & Εγγραφή Πληροφοριών

Η συνάρτηση **printf()** όταν χρησιμοποιείται θα πρέπει να συμπεριλάβει και το αρχείο επικεφαλίδας **stdio.h**

Συντάσσεται ως εξής:

printf (format string [ορίσματα,...])

Όπου **format string** ακολουθία χαρακτήρων μορφής

Η ακολουθία χαρακτήρων μορφής της **printf()** καθορίζει την μορφοποίηση της εξόδου.

Ανάγνωση & Εγγραφή Πληροφοριών

Τα συστατικά της ακολουθίας χαρακτήρων μορφής είναι:

- * Κυριολεκτικό κείμενο
- * Αλληλουχία διαφυγής (\n)
- * Προσδιοριστικό μετατροπής (%)

Αλληλουχία διαφυγής

\t οριζόντιο tab
\n νέα γραμμή

Προσδιοριστικό μετατροπής

%c	char
%d	int, short
%f	float, double
%s	char array

Ανάγνωση & Εγγραφή Πληροφοριών

Η συνάρτηση **puts()**

- ❖ προβάλλει μόνο μηνύματα κειμένου στην οθόνη
- ❖ δεν προβάλλει τιμές αριθμητικών μεταβλητών
- ❖ Μπορεί να περιέχει αλληλουχίες διαφυγής

Η είσοδος και η έξοδος των πληροφοριών γίνεται με **ροές** ανεξάρτητα την προέλευση της εισόδου ή την κατεύθυνση της εξόδου

Διαχείριση των Ροών για Είσοδο/Έξοδο

Ροή είναι μια αλληλουχία χαρακτήρων. Ακριβέστερα είναι μια αλληλουχία από bytes δεδομένων

- ❖ **Ροές** εισόδου & **Ροές** εξόδου

Κάθε **Ροή** συνδέεται σε ένα αρχείο.

- ❖ **Ροές** κειμένου & **Ροές** δυαδικές
- ❖ Οι **Ροές** κειμένου οργανώνονται σε γραμμές έως 255 χαρακτήρες & τερματίζουν από χαρακτήρες τέλους ή νέας γραμμής
- ❖ Οι δυαδικές **Ροές** χρησιμοποιούνται με αρχεία δίσκου.

Διαχείριση των Ροών για Είσοδο/Έξοδο

Οι Ροές

ανοίγουν αυτομάτως όταν ένα πρόγραμμα ξεκινά την εκτέλεσή του και

κλείνουν όταν το πρόγραμμα τερματίζει

Όνομα	Ροές	Συσκευή
stdin	Τυπική είσοδος	Πληκτρολόγιο
stdout	Τυπική έξοδος	Οθόνη
stderr	Τυπική σφάλμα	Οθόνη
stderrn	Τυπική εκτυπωτής	Εκτυπωτής
stdaux	Τυπική βοηθητική	Σειριακή θύρα

Διαχείριση των Ροών για Είσοδο/Έξοδο

Η τυπική βιβλιοθήκη της C έχει ποικιλία συναρτήσεων για την διαχείριση των **Ροών**.

Τυπικές Ροές	Όνομα Ροής	Περιγραφή
printf()	fprintf()	Μορφοποιημένη έξοδος
vprintf()	vfprintf()	Μορφοποιημένη έξοδος με μεταβλητή λίστα ορισμάτων
puts()	fputs()	Έξοδος ακολουθίας χαρακτήρων
putchar()	putc(), fputc()	Έξοδος χαρακτήρων
scanf()	fscanf()	Μορφοποιημένη είσοδος
vscanf()	vfscanf()	Μορφοποιημένη είσοδος με μεταβλητή λίστα ορισμάτων
gets()	fgets()	Είσοδος ακολουθίας χαρακτήρων
getchar()	getc(), fgetc()	Είσοδος χαρακτήρων
perror()		Έξοδος ακολουθίας χαρακτήρων μόνο στην stderr

Διαχείριση των Ροών για Είσοδο/Εξοδο

Η συνάρτηση `getchar()`

`int getchar(void);`

Η συνάρτηση παίρνει τον επόμενο χαρακτήρα από την ροή **`stdin`**. Παρέχει **`buffered`** είσοδο δηλ. οι χαρακτήρες δεν λαμβάνονται αν δεν πατήσεις **`enter`**

Η συνάρτηση `getchar()` μπορεί να χρησιμοποιηθεί για την είσοδο ολόκληρων γραμμών κειμένου.

Διαχείριση των Ροών για Είσοδο/Εξοδο

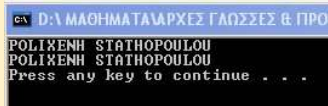
```
#include <stdio.h>
#include <stdlib.h>

int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getchar()) != '\n')
        putchar(ch);
    system("PAUSE");
    return 0;
}
```

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

int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getchar()) != '\n')
        printf("%c",ch);
    printf("\n"); system("PAUSE");

    return 0;
}
```



```
D:\ΜΑΘΗΜΑΤΑ\ΑΡΧΕΣ ΓΛΩΣΣΕΣ & ΠΡΟ
ΡΟΛΙΩΝΗ ΣΤΑΘΟΡΟΥΛΟΥ
ΡΟΛΙΩΝΗ ΣΤΑΘΟΡΟΥΛΟΥ
Press any key to continue . . .
```


Διαχείριση των Ροών για Είσοδο/Εξοδο

```
* Using getchar() to input strings. */
#include <stdio.h>
#define MAX 80
int main( void )
{
    char ch, buffer[MAX+1];
    int x = 0;
    while ((ch = getchar()) != '\n' && x < MAX)
        buffer[x++] = ch;
    buffer[x] = '\0';
    printf("%s\n", buffer);

    return 0;
}
```

Διαχείριση των Ροών για Είσοδο/Εξοδο

Η συνάρτηση `getch()`

`int getch(void);`

Η συνάρτηση παίρνει τον επόμενο χαρακτήρα από την ροή **stdin**. Παρέχει **unbuffered** είσοδο δηλ. επιστρέφει κάθε χαρακτήρα μόλις πατηθεί το πλήκτρο χωρίς να περιμένει το **enter**. Δεν είναι εντολή του πρότυπου ANSI.

Οι χαρακτήρες δε προβάλλονται στην Οθόνη

Απαιτεί διαφορετικά αρχεία επικεφαλίδας `<conio.h>`

Διαχείριση των Ροών για Είσοδο/Εξοδο

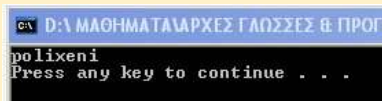
```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getch()) != '\r')
        putchar(ch);
    system("PAUSE");
    return 0;
}
```

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

int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getch()) != '\r')
        printf("%c",ch);
    printf("\n"); system("PAUSE");

    return 0;
}
```



Διαχείριση των Ροών για Είσοδο/Εξοδο

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

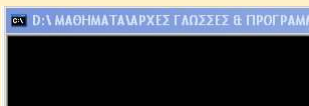
int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getch()) != '\r');

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

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

int main(int argc, char *argv[])
{
    int ch;
    while ((ch = getch()) != '\r');

    printf("\n");2
    system("PAUSE");
    return 0;
}
```



Διαχείριση των Ροών για Είσοδο/Έξοδο

Η συνάρτηση **getche()** αντηχεί κάθε χαρακτήρα στην **stdout**. Δεν είναι εντολή του πρότυπου ANSI.

Οι χαρακτήρες προβάλλονται στην Οθόνη δύο φορές

Οι συναρτήσεις **getc()** & **fgetc()** δεν λειτουργούν αυτομάτως με την **stdin**. Χρησιμοποιούνται κυρίως για να διαβάζουν χαρακτήρες από αρχεία.

Εμβέλεια Μεταβλητών

Η **εμβέλεια** μιας μεταβλητής σημαίνει **πρόσβαση** στη μεταβλητή δηλ. (**Ορατή** η μεταβλητή) από διαφορετικά τμήματα του προγράμματος

προσβασιμότητα ή **ορατότητα** στις μεταβλητές

Η **εμβέλεια** αφορά όλους τους τύπους μεταβλητών

- | | |
|-----------------------|------------|
| * απλές μεταβλητές, | * πίνακες, |
| * δομές, | * δείκτες, |
| * συμβολικές σταθερές | |

Εμβέλεια Μεταβλητών

```
#include <stdio.h>
int x = 999;
void print_value(void);
int main(int argc, char *argv[])
{
    printf("%d\n", x);
    print_value();
    system("PAUSE");
    return 0;
}
void print_value(void)
{
    printf("%d\n", x);
}
```

Η μεταβλητή x είναι προσπελάσιμη παντού

Εμβέλεια Μεταβλητών

```
#include <stdio.h>
void print_value(void);
int main(int argc, char *argv[])
{
    int x = 999;
    printf("%d\n", x);
    print_value();
    system("PAUSE");
    return 0;
}
void print_value(void)
{
    printf("%d\n", x);
}
```

Η μεταβλητή **x** είναι προσπελάσιμη μόνο στο κυρίως πρόγραμμα

Κατά την μεταγλώττιση θα εμφανίσει λάθος

Εμβέλεια Μεταβλητών

Η εμβέλεια των εξωτερικών μεταβλητών εκτείνεται σε ολόκληρο το πρόγραμμα

καλή πρακτική να δηλώνουμε
Τις εξωτερικές μεταβλητές

Πως?

`extern type NameVariable;`

```
#include <stdio.h>
int x = 999;
void print_value(void);
int main( void )
{
    extern int x;
    printf("%d\n", x);
    print_value();
    return 0;
}

void print_value(void)
{
    extern int x;
    printf("%d\n", x);
}
```

Εμβέλεια Μεταβλητών

Η εμβέλεια των τοπικών μεταβλητών περιορίζεται στις συναρτήσεις που ορίζονται

αυτόματες τοπικές
μεταβλητές

στατικές τοπικές
μεταβλητές

Οι αυτόματες
δημιουργούνται
κάθε φορά

Οι στατικές διατηρούνται
ανάμεσα στις διαδοχικές
κλήσεις

Εμβέλεια Μεταβλητών

```
/* Demonstrates automatic and static local variables. */
#include <stdio.h>
void func1(void);
int main( void )
{
    int count;

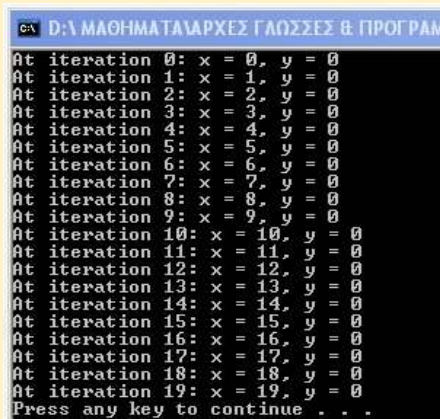
    for (count = 0; count < 20; count++)
    {
        printf("At iteration %d: ", count);
        func1();
    }

    return 0;
}

void func1(void)
{
    static int x = 0;
    int y = 0;

    printf("x = %d, y = %d\n", x++, y++);
}
```

Εμβέλεια Μεταβλητών



```
C:\ D:\ ΜΑΘΗΜΑΤΑ\ΑΡΧΕΣ ΓΛΩΣΣΕΣ & ΠΡΟΓΡΑΜ
At iteration 0: x = 0, y = 0
At iteration 1: x = 1, y = 0
At iteration 2: x = 2, y = 0
At iteration 3: x = 3, y = 0
At iteration 4: x = 4, y = 0
At iteration 5: x = 5, y = 0
At iteration 6: x = 6, y = 0
At iteration 7: x = 7, y = 0
At iteration 8: x = 8, y = 0
At iteration 9: x = 9, y = 0
At iteration 10: x = 10, y = 0
At iteration 11: x = 11, y = 0
At iteration 12: x = 12, y = 0
At iteration 13: x = 13, y = 0
At iteration 14: x = 14, y = 0
At iteration 15: x = 15, y = 0
At iteration 16: x = 16, y = 0
At iteration 17: x = 17, y = 0
At iteration 18: x = 18, y = 0
At iteration 19: x = 19, y = 0
Press any key to continue . . .
```