



Κεφάλαιο 5.4-5.11:

# Επαναλήψεις (οι βρόχοι **for**, **do-while**)

(Διάλεξη 10)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

# Εντολές Επανάληψης που θα καλυφθούν σήμερα



## Διάλεξη 9 - Δευτέρα

- `while(){}`
- τελεστές postfix/prefix (`++`, `--`, ...) και σύνθετοι τελεστές
- Παραδείγματα

## Διάλεξη 10 - Σήμερα

- `for(){}`
- Η εντολές `break/continue`;
- `do{ }while()`
- Φωλιασμένοι Βρόχοι (Nested Loops)
- Παραδείγματα

# Σύνταξη του Βρόχου **for**



```
for(αρχικοποίηση; συνθήκη επανάληψης ; ενημέρωση)  
    εντολή;
```

```
for(αρχικοποίηση; συνθήκη επανάληψης; ενημέρωση) {  
    εντολη;  
    εντολη;  
    .....  
}
```

**Προσοχή: Γιατί είναι λάθος η πιο κάτω πρόταση**

```
for (αρχικοποίηση; συνθήκη επανάληψης ; ενημέρωση);  
    εντολή;
```

# Παράδειγμα με for



```
int x;  
for (x=0; x<5; x++) {  
    printf( "%d\n" , x );  
}
```

x	x<5	έξοδος
0	1	0
1	1	1
2	1	2
3	1	3
4	1	4
5	0	

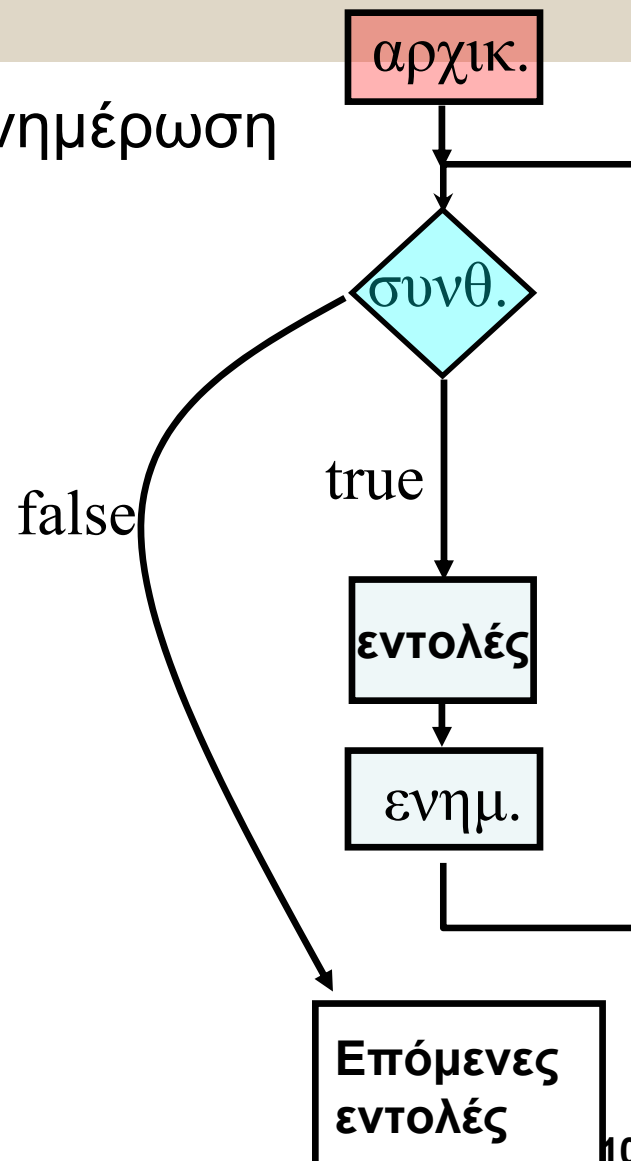


# Ροή Έλεγχου for

```
int x,y;  
for (x=0; x<5; x++) {  
    printf("%d\n",x);  
}  
// Εδώ το x==5  
y = x*x + 10;  
printf("%d\n",y);  
// επομένως εκτυπώνεται 35
```

1) Συνθήκη      3) Ενημέρωση

2) Εντολές



# Ομοιότητα for με while



```
int x = 0;
while (x < 5) {
    printf("%d\n",x);
    x++;
}
```

```
int x;
for (x=0; x<5; x++) {
    printf("%d\n",x);
}
```

**Οποιοδήποτε for μπορεί να γραφεί με while και οποιοδήποτε while με for ακόμα και το άπειρο loop**

```
for (;1;) {
    printf("Hello");
}
```



```
while (1)
    printf("Hello");
}
```

# Παραδείγματα με for



**Μπορούμε να μετρήσουμε και αντίστροφα**

```
int x;
for (x=5; x>0; x--){
    printf("%d\n",x)
}
ΕΚΤΥΠΩΝΕΙ 5,4,3,2,1,
```

**Ή με μεγαλύτερο βήμα (step increment) ή ότι επιθυμούμε**

```
int i;
for (i = 0; i < 10; i += 2) {
    printf ("%d ", i);
}
ΕΚΤΥΠΩΝΕΙ 0,2,4,6,8,
```

```
int i = 10;
for (i = 10; i > 0; i /= 2) {
    printf ("%d ",i);
}
ΕΚΤΥΠΩΝΕΙ: 10,5,2,1
```

# Η Εντολή do-while



- Στις εντολές for και while, προτού εκτελεστεί κάποια εντολή στο body, πρέπει να ελεγχθεί κάποια συνθήκη.  
π.χ. `int a =0; while (a<2) { /* body */ a = a+1; }`  
π.χ. `int for (a=0;a<2;a++) { /* body */ }`
- Σε αρκετές περιπτώσεις αυτό δεν είναι βολικό διότι δεν υπάρχει η τιμή της μεταβλητής στην συνθήκη.
- Παράδειγμα ακολουθεί...



# Η Εντολή do-while



## Πρόβλημα

Θέλουμε να διαβάσουμε αριθμούς από τον χρήστη και να τους αθροίσουμε. Όταν ο χρήστης δώσει 0 τότε τερματίζει το πρόγραμμα.

Με **While**

```
scanf("%d", &a);  
while (a!=0) {  
    sum+=a;  
    scanf("%d", &a);  
}
```

Με **For**

```
scanf("%d", &a);  
for (;a!=0;) {  
    sum+=a;  
    scanf("%d", &a);  
}
```

**Πρόβλημα:** Και στις δυο περιπτώσεις χρειαζόμαστε 2 scanf

# Η Εντολή do-while



```
do
    εντολή;
while(συνθήκη);
```

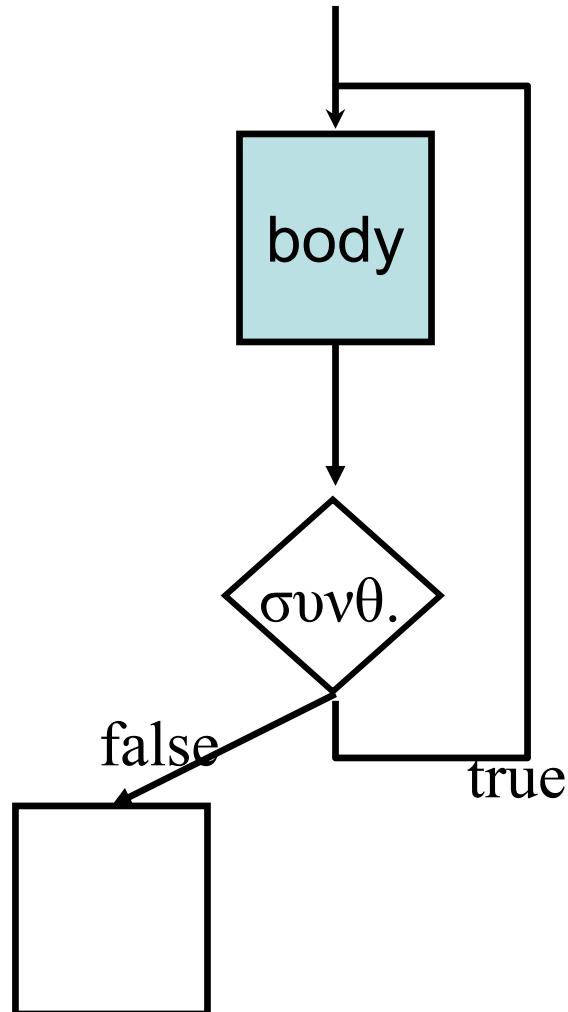
```
do{
    εντολή;
    εντολή;
}while(συνθήκη);
```

```
do {
    scanf("%d", &a);
    sum+=a;
} while (a!=0);
```

Πρώτα εκτελείται το σώμα της `do { /* σώμα */ } while();`

Ο έλεγχος της συνθήκης γίνεται μετά το `while`

# Ροη Ελεγχου do-while





# Παράδειγμα **break** σε βρόχο

Η εντολή **break** μπορεί να χρησιμοποιηθεί σε βρόχο **while**, **do-while** και **for** για να προκαλέσει έξοδο από το βρόχο.

```
#include <stdio.h>
main ( )
{
    int i;
    for (i = 1; i < 10; i++)
    {
        if (i == 5)
        {
            break;
        }
        printf ("%d ", i);
    }
    printf ("\nbroke out of loop at i = %d\n", i);
}
```

**ΈΞΟΔΟΣ:**

**1 2 3 4**

**Έξοδος από βρόχο με i = 5**



# Παράδειγμα **continue** σε βρόχο

Η εντολή **continue** μπορεί να χρησιμοποιηθεί με **for**, **while**, **do-while** για να προκαλέσει την ματαίωση της εκτέλεσης των επομένων εντολών του βρόχου, για την τρέχουσα επανάληψη.

```
#include <stdio.h>
main ( )
{
    int i;
    for (i = 1; i < 10; i++)
    {
        if (i == 5)
        {
            continue;
        }
        printf ("%d", i);
    }
    printf ("\n");
}
```

**OUTPUT:**

**1 2 3 4 6 7 8 9**

# Πρόβλημα Εύρεσης Παραγοντικού



Σχεδιάστε αλγόριθμο ο οποίος υπολογίζει το παραγοντικό ενός ακέραιου αριθμού που δίνει ο χρήστης.

- Δεδομένα εισόδου: αριθμός  $x$
- Δεδομένα εξόδου : το παραγοντικό

Π.χ.  $1! = 1$ ,  $2! = 1.2 = 2$ ,  $3! = 1.2.3 = 6$   
 $4! = 1.2.3.4 = 24$ ,  $5! = 1.2.3.4.5 = 120$ , ....

# Λύση 1: Εύρεσης Παραγοντικού με for loop



```
int main()
{
    int i, factorial=1;
    int n;

    printf("Enter Number (1-12):");
    scanf("%d", &n);

    for (i=1; i<=n; i++) {
        factorial *= i;
    }

    printf("The factorial of %d is %d\n", n, factorial);
    return 0;
}
```

# Λύση 2: Εύρεσης Παραγοντικού με έλεγχο δεδομένων εισόδου



```
int main()
{
    int i, factorial=1;
    int n;

    do { // Έλεγχος ότι τα δεδομένα εισόδου είναι ορθά
        printf("Enter Number (1-12):");
        scanf("%d", &n);
    } while (n<1 || n>12);

    for (i=1; i<=n; i++) {
        factorial *= i;
    }
    printf("The factorial of %d is %d\n", n, factorial);
    return 0;
}
```

Ελέγχουμε ότι η τιμή είναι μεταξύ 1-12



# Λύση 3: Εύρεσης Παραγοντικού με **while** loop



```
int main()
{
    int i=1, factorial=1;
    int n;

    do {
        // Έλεγχος ότι τα δεδομένα εισόδου είναι ορθά
        printf("Enter Number (1-12):");
        scanf("%d", &n);
    } while (n<1 || n>12);

    while (i<=n) {
        factorial *= i;
        i++;
    }

    printf("The factorial of %d is %d\n", n, factorial);
    return 0;
}
```

Η αρχικοποίηση γίνεται εδώ

Η εκτέλεση με την χρήση while loop

# Λύση 3: Εύρεσης Παραγοντικού με **do while** loop



```
int main()
{
    int i=1, factorial=1;
    int n;

    do {
        // Έλεγχος ότι τα δεδομένα εισόδου είναι ορθά
        printf("Enter Number (1-12):");
        scanf("%d", &n);
    } while (n<1 || n>12);

    do {
        factorial *= i;
        i++;
    } while (i<=n)

    printf("The factorial of %d is %d\n", n, factorial);
    return 0;
}
```

→ Η αρχικοποίηση γίνεται εδώ

→ Η εκτέλεση με χρήση do-while loop

# Φωλιασμένοι (Nested) Βρόχοι :

## Παράδειγμα 1



Σε ένα πρόγραμμα μπορούμε να φωλιάσουμε ένα βρόχο μέσα σε ένα άλλο βρόχο. Πχ.

```
int i=0, j;
while(i<5){
    j=0;
    while(j<5){
        printf(“%d-%d”,i,j);
        j++;
    }
    printf(“\n”);
    i++;
}
```

### Αποτέλεσμα

```
0-0 0-1 0-2 0-3 0-4
1-0 1-1 1-2 1-3 1-4
2-0 2-1 2-2 2-3 2-4
3-0 3-1 3-2 3-3 3-4
4-0 4-1 4-2 4-3 4-4
```

# Φωλιασμένοι (Nested) Βρόχοι :

## Παράδειγμα 2



Ας κτίσουμε ένα τρίγωνο...

```
int i=0,j;
while(i<5){
    j=0;
    while(j<=i){
        printf(“%d-%d”,i,j);
        j++;
    }
    printf(“\n”);
    i++;
}
```

**Αποτέλεσμα**

0-0

1-0 1-1

2-0 2-1 2-2

3-0 3-1 3-2 3-3

4-0 4-1 4-2 4-3 4-4