



Κεφάλαιο 5.1-5.3:

Επαναλήψεις (οι βρόχος While)

(Διάλεξη 9)

Διδάσκων: Δημήτρης Ζεϊναλιπούρ

Δομές Έλεγχου Προγράμματος



- Μέχρι τώρα είδαμε τις ακόλουθες δομές έλεγχου (*program control structure*) ενός προγράμματος
 - A) Sequence** π.χ. `int a=1; b=34; printf("Hello");`
 - B) Selection** π.χ. `If (a>1) { printf("a>1"); }` ή `switch (a) { }`
- Σήμερα θα δούμε ένα τρίτο τύπο δομής έλεγχου ενός προγράμματος:
 - Γ) Repetition (Επαναληπτική Δομή Έλεγχου)**
π.χ. `while (a>0) { }` ή `for (i=0;i<5;i++) { }`
- Η δομές επανάληψης επιτρέπουν στους προγραμματιστές να προσδιορίσουν μία **ενέργεια** που θα **επαναλαμβάνεται** όσο μία **συνθήκη** παραμένει αληθής.

Εισαγωγή: Δομές Επανάληψης



Παράδειγμα

Θέλουμε να τυπώσουμε τους αριθμούς από το μηδέν έως τρία στην οθόνη

Ανεπιτυχής Προσπάθεια

```
int main() {  
    printf("0");  
    printf("1");  
    printf("2");  
    printf("3");  
    return 0;  
}
```

Τι γίνεται αν θέλω να εκτυπώσω τους αριθμούς από 0 έως 100?

Φαίνεται ότι η ρητή απαρίθμηση δεν είναι πολύ εύχρηστη

Εισαγωγή: Δομές Επανάληψης



Επιτυχής Προσπάθεια

```
int a = 0;
```

```
while (a < 4) {
```

```
    printf("%d, ", a);
```

```
    a = a + 1;
```

```
}
```

→ συνθήκη

} → Ενέργεια
επανάληψης

(iteration)

Μας τυπώνει 0, 1, 2, 3,

Εισαγωγή: Δομές Επανάληψης



Ας δούμε τι ακριβώς εκτελείτε

α	$\alpha < 4$	output
0	1	0
1	1	1
2	1	2
3	1	3
4	0	

```
int a=0;
while ( $\alpha < 4$ ) {
    printf("%d, ", a);
    a = a+1;
}
```

Εισαγωγή: Δομές Επανάληψης



Εκτύπωση σε αντίστροφη σειρά

```
int a = 4;
```

```
while (a>0) {  
    a = a-1;  
    printf("%d, ", a);  
}
```

Μας τυπώνει 3,2,1,0

Σιγουρευτείτε ότι καταλαβαίνεται την διαφορά του while και if

```
int a = 4;
```

```
if (a>0) {  
    a = a-1;  
    printf("%d, ", a);  
}
```

Μας τυπώνει 3

Κατηγορίες Βρόχων Επανάληψης

- Για δημιουργία επανάληψης σε ένα πρόγραμμα υπάρχουν διάφοροι τρόποι:

A) Counting Loops (Με μετρητή)

Προσδιορισμένος αριθμός επαναλήψεων

π.χ. `int a =0; while (a<2) { printf("Hello"); a = a+1; }`

B) Sentinel Loops (Με σημαία)

Απροσδιόριστος αριθμός επαναλήψεων. Το loop διακόπτεται όταν δοθεί κάποιος ειδικός χαρακτήρας.

π.χ. `int a =0; while (a!=-1) { scanf("%d", &a); printf("Hello"); }`

C) Συνδυασμός (με χρήση λογικών τελεστών)

Εντολές Επανάληψης που θα καλυφθούν αυτή την βδομάδα



Διάλεξη 9 - Σήμερα

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

Διάλεξη 10 - Πέμπτη

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

While Loop



Σύνταξη:

```
while (έκφραση)  
    εντολή;
```

```
while (έκφραση) {  
    εντολη1;  
    εντολη2;  
    ...  
}
```

Σημασία:

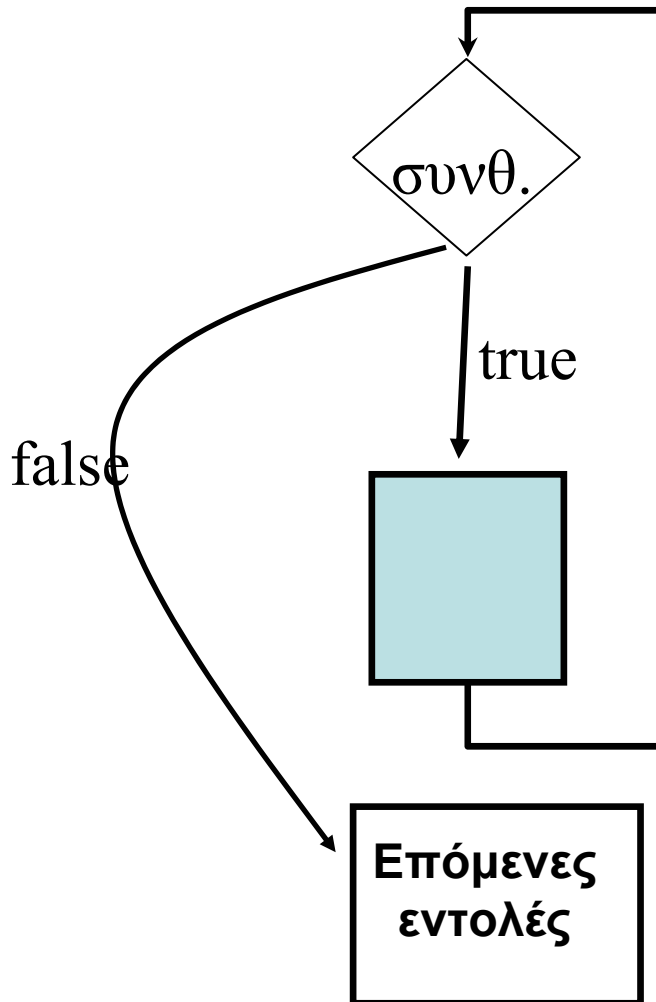
εφόσον η έκφραση (συνθήκης == αληθής) εκτέλεσε τις εξαρτώμενες εντολές *αλλιώς συνέχισε με τις εντολές που ακολουθούν το while block*

ΠΡΟΣΟΧΗ ΛΑΘΟΣ!!!

```
while (εκφραση);  
    εντολη;
```

↙

Ροή Ελέγχου while



εφόσον η έκφραση (συνθήκης == αληθής) εκτέλεσε τις εξαρτώμενες εντολές *αλλιώς συνέχισε με τις εντολές που ακολουθούν το while block*

Πρόβλημα 1: Εκτύπωση Αριθμών



Γράψετε ένα τμήμα προγράμματος που τυπώνει τους ακέραιους αριθμούς από το **1 μέχρι το 12**.

Κάθε αριθμός που **διαίρεείται με το 3** να τυπώνεται * δίπλα του. Ο κάθε αριθμός να τυπώνεται σε **ξεχωριστή γραμμή**.

Δηλαδή:

1

2

3*

4

5

6*

...

Πρόβλημα 1: Λύση (Εκτύπωση Αριθμών)



```
int i;    /* loop counter */
i = 1;
while (i <= 12) {
    printf("%d", i);
    if((i%3) == 0)
    {
        printf("*");
    }
    printf("\n");
    i = i + 1;
}
```

Τελεστές *Prefix and Postfix*



Συχνά θέλουμε να αυξάνουμε/μειώνουμε μετρητές κατά 1. Υπάρχει πιο εύκολος τρόπος!

- Prefix Operator (*προ-σημειογραφική*)

`++i;` $\langle == \rangle$ `i = i + 1;`

`--i;` $\langle == \rangle$ `i = i - 1;`

- Postfix Operator (*μετά-σημειογραφική*)

`i++;` $\langle == \rangle$ `i = i + 1;`

`i--;` $\langle == \rangle$ `i = i - 1;`

Και οι δυο τελεστές εκτελούν την ίδια πράξη (αφαίρεση / πρόσθεση) αλλά με διαφορετική σειρά.

Παραδείγματα ακολουθούν...

Prefix vs Postfix Τελεστές



`i = 5;`
`x = ++i;`
`y = i++;`

→ Πρώτα αύξησε και μετά ανάθεσε
x=6

→ Πρώτα ανάθεσε και μετά αύξησε
y=6
i=7

- x είναι 6, y είναι 6 και το i είναι 7

```
int my_int = 3;
printf ("%d, ", my_int++);
printf ("%d, ", my_int++);
printf ("%d, ", my_int);
ΕΚΤΥΠΩΝΕΙ: 3,4,5
```

```
int my_int = 3;
printf ("%d, ", ++my_int);
printf ("%d, ", ++my_int);
printf ("%d, ", my_int);
ΕΚΤΥΠΩΝΕΙ: 4,5,5
```

Σύνθετοι Τελεστές Ανάθεσης



Συχνά θέλουμε να αυξάνουμε / μειώνουμε μετρητές κατά κάποιο $K (>1)$ Υπάρχει πιο συνοπτικός τρόπος!

- Πρόσθεση: $i+=k;$ $\iff i = i + k;$
- Πολλαπλασιασμός: $i*=k;$ $\iff i = i * k;$
- Διαίρεση: $i/=k;$ $\iff i = i / k;$
- Υπόλοιπο: $i\%=k;$ $\iff i = i\%k;$

$$i\mathit{op}= k; \iff i = i \mathit{op} (k);$$

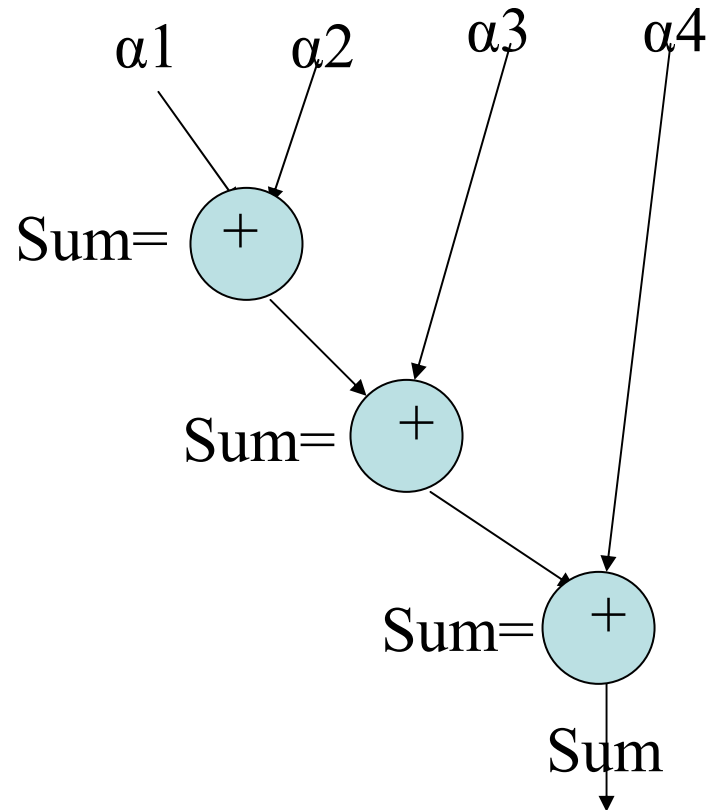
Πρόβλημα: Άθροισμα Σειράς - Περιγραφή

- Γράψετε ένα πρόγραμμα το οποίο υπολογίζει το άθροισμα μιας σειράς ακέραιων αριθμών.
- Η σειρά εισάγεται από μονάδα εισόδου και τερματίζεται με την τιμή 0.
- π.χ.

```
Enter series: 2 4 236 6 31 23 0  
>> 302
```


Πρόβλημα Άθροισμα Σειράς:

b) Άθροιση Στοιχείων



Πρόβλημα Άθροισμα Σειράς: Ολοκληρωμένος κώδικας



```
int number;    /* holds input number one at a time */
int sum=0;     /* current sum */

// διάβασμα πρώτου αριθμού
scanf("%d",&number);

while(number != 0){
    // εκτέλεση πράξης
    sum += number; // άθροιση με υπάρχων sum

    // διάβασμα επόμενου
    scanf("%d",&number);
}

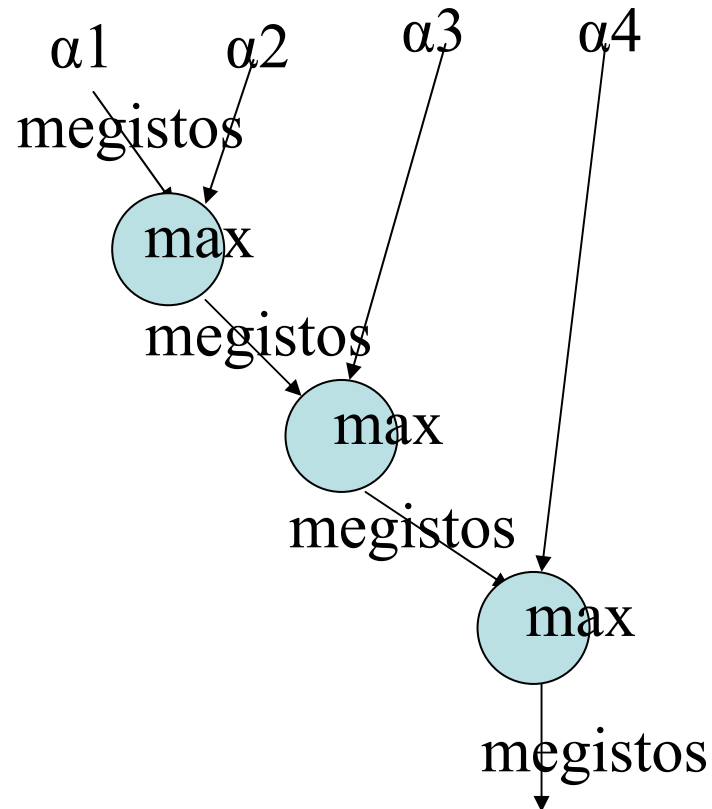
printf("Το άθροισμα της σειράς είναι %d\n",sum);
```

Πρόβλημα Εύρεσης Μέγιστου Σειράς



- Γράψετε ένα πρόγραμμα που αναγνωρίζει τον πιο μεγάλο αριθμό σε μια σειρά ακέραιων αριθμών. Η σειρά εισάγεται από μονάδα εισόδου και τερματίζεται με την τιμή 0.
- Π.χ.
Enter series: 2 4 5236 6 31 978 5555 23 0
>> 5555

Πρόβλημα Εύρεσης Μέγιστου από σειρά N στοιχείων



Πρόβλημα Εύρεσης Μέγιστου Σειράς: Ολοκληρωμένος Κώδικας



```
int main()
{
    int number;                /* hold input number one at a time */
    int maximum;              /* hold current maximum */

    scanf("%d",&number);      /* diabase prwto stoixeio */
    maximum = number;         /* arxikopoihsh */

    while(number != 0){
        scanf("%d",&number); /* διαβασε επομενο στοιχειο */
        if (number > maximum)
            maximum = number;
    }
    printf("To megisto stoixeio της σειρας είναι %d\n", maximum);
}
```