

# Διάλεξη 5η: Εντολές Επανάληψης

Τμήμα Επιστήμης Υπολογιστών, Πανεπιστήμιο Κρήτης

Εισαγωγή στην Επιστήμη Υπολογιστών

Βασίζεται σε διαφάνειες του Κ. Παναγιωτάκη



# Είδη εντολών

- Σειριακές (sequential) εντολές, εκτελείται η μία μετά την άλλη
  - `x = 42; z = f(x); ...`
- Εντολές απόφασης (conditional), αποφασίζουν τι θα εκτελεστεί, υπό συνθήκη
  - `if (x == 0) { y = 1; } else { y = 2; }`
  - `switch (x) { case 1: ... case 2: ... default: ... }`
- Εντολές επανάληψης, εκτελούν πολλές φορές την ίδια λειτουργία
  - `for (i = 0; i < 10; i++) { ... }`
  - `while (f(x) < 2) { ... }`
  - Μια εντολή επανάληψης λέγεται και βρόχος γιατί “κυκλώνει” άλλες εντολές



# Επαναληπτικές εντολές: `while`

- Εντολή `while`: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
while (expression) {  
    statements  
}
```



# Επαναληπτικές εντολές: `while`

- Εντολή `while`: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
while (expression) {  
    statements  
}
```

Υπολογίζεται η τιμή της έκφρασης (expression) ως ακέραιος αριθμός, και συγκρίνεται με το 0



# Επαναληπτικές εντολές: while

- Εντολή **while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
while (expression) {  
    statements  
}
```

Εντολές (statements) που εκτελούνται μόνο αν η έκφραση έχει τιμή διαφορετική από 0



# Επαναληπτικές εντολές: `while`

- Εντολή `while`: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
while (expression) {  
    statements  
}
```

Υπολογίζεται ξανά η τιμή της έκφρασης (expression) ως ακέραιος αριθμός, και συγκρίνεται με το 0



# Επαναληπτικές εντολές: while

- Εντολή **while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
while (expression) {  
    statements  
}
```

Αν η έκφραση έχει τιμή  
διαφορετική από 0,  
εκτελούνται ξανά οι εντολές,  
κλπ



# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος



# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Έξοδος

Αρχικά, το **c** είναι 1, και το **sum** είναι 0

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

`while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Ισχύει το  $1 \leq 10$ , οπότε θα  
εκτελεστούν οι εντολές του  
`while`

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

c = 1, c\*c = 1

Εκτυπώνει το  
μήνυμα για c ίσο με  
1

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Έξοδος

c = 1, c\*c = 1

Υπολογίζει το  $c*c$  και το  
προσθέτει στο  $sum$

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c*c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Αυξάνει το c κατά 1

## Έξοδος

c = 1, c\*c = 1

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Επανάλεγχος της συνθήκης:  
Ισχύει το  $2 \leq 10$ , οπότε θα  
εκτελεστούν ξανά οι εντολές  
του **while**

Έξοδος

c = 1, c\*c = 1

c = 2, c\*c = 4

c = 3, c\*c = 9

c = 4, c\*c = 16

c = 5, c\*c = 25

c = 6, c\*c = 36

c = 7, c\*c = 49

c = 8, c\*c = 64

c = 9, c\*c = 81

c = 10, c\*c = 100

1\*1 + 2\*2 + ... + 10\*10 = 385

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

c = 1, c\*c = 1  
c = 2, c\*c = 4  
Εκτυπώνει το  
μήνυμα για c ίσο με  
2

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
```

Υπολογίζει το  $c*c$  και το προσθέτει στο  $sum$



# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c*c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Αυξάνει το **c** κατά 1

## Έξοδος

c = 1, c\*c = 1  
c = 2, c\*c = 4

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Επανάλεγχος της  
συνθήκης: Ισχύει το  $3 \leq 10$ , οπότε θα  
εκτελεστούν ξανά οι  
εντολές του **while**

Έξοδος

c = 1, c\*c = 1  
c = 2, c\*c = 4

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
```



# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
c = 9, c*c = 81
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
c = 9, c*c = 81
c = 10, c*c = 100
```

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

Επανελέγχος της συνθήκης:  
Δεν ισχύει το  $11 \leq 10$ , οπότε  
τελειώνει η εκτέλεση της

while

Εξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
c = 9, c*c = 81
c = 10, c*c = 100
```

# Παράδειγμα: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## `while.c`

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
c = 9, c*c = 81
c = 10, c*c = 100
```

Η εκτέλεση  
συνεχίζει με την  
επόμενη εντολή

# Παράδειγμα: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10

## while.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0;

    while (c <= 10) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
        c++;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

## Έξοδος

```
c = 1, c*c = 1
c = 2, c*c = 4
c = 3, c*c = 9
c = 4, c*c = 16
c = 5, c*c = 25
c = 6, c*c = 36
c = 7, c*c = 49
c = 8, c*c = 64
c = 9, c*c = 81
c = 10, c*c = 100
```

```
1*1 + 2*2 + ... +
10*10 = 385
```

## Παράδειγμα 2: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### `while2.c`

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

## Παράδειγμα 2: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### `while2.c`

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

## Παράδειγμα 2: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### `while2.c`

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81



## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

## Παράδειγμα 2: `while`

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### `while2.c`

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

`c = 10, c*c = 100`

`c = 9, c*c = 81`

`c = 8, c*c = 64`

`c = 7, c*c = 49`

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

c = 4, c\*c = 16

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

c = 4, c\*c = 16

c = 3, c\*c = 9

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

c = 4, c\*c = 16

c = 3, c\*c = 9

c = 2, c\*c = 4

## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

c = 4, c\*c = 16

c = 3, c\*c = 9

c = 2, c\*c = 4

c = 1, c\*c = 1



## Παράδειγμα 2: while

- Άθροισμα των τετραγώνων των αριθμών από 1 έως 10, εναλλακτικός τρόπος

### while2.c

```
#include <stdio.h>
int main()
{
    int c = 11, sum = 0;

    while (--c) {
        printf("c = %d, c*c = %d\n", c, c * c);
        sum += c * c;
    }
    printf("1*1 + 2*2 + ... + 10*10 = %d\n", sum);
    return 0;
}
```

### Έξοδος

c = 10, c\*c = 100

c = 9, c\*c = 81

c = 8, c\*c = 64

c = 7, c\*c = 49

c = 6, c\*c = 36

c = 5, c\*c = 25

c = 4, c\*c = 16

c = 3, c\*c = 9

c = 2, c\*c = 4

c = 1, c\*c = 1

1\*1 + 2\*2 + ... +  
10\*10 = 385

# Παράδειγμα 3: while

- Άθροισμα των τετραγώνων των 10 πρώτων αριθμών που διαιρούνται με 7 ή 9.

## while3.c

```
#include <stdio.h>
int main()
{
    int c = 1, sum = 0, pl = 0;

    while (pl < 10) {
        if ((c % 7) == 0 || (c % 9) == 0) {
            printf("c = %d, c*c = %d\n", c, c * c);
            sum += c * c;
            pl++;
        }
        c++;
    }
    printf("Sum = %d\n", sum);
    return 0;
}
```



# Επαναληπτικές εντολές: **for**

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές N φορές.

## Γενική μορφή

```
for (expression1; expression2; expression3 ) {  
    statements;  
}
```



# Επαναληπτικές εντολές: for

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές  $N$  φορές.

## Γενική μορφή

```
for (expression1; expression2; expression3) {  
    statements;  
}
```

Εκτελείται αρχικά η  
έκφραση **expression1**



# Επαναληπτικές εντολές: for

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές N φορές.

Υπολογίζεται η τιμή της έκφρασης **expression2**

## Γενική μορφή

```
for (expression1; expression2; expression3 ) {  
    statements;  
}
```



# Επαναληπτικές εντολές: for

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές N φορές.

## Γενική μορφή

```
for (expression;  
statements;  
)
```

Αν είναι αληθής (μη  
μηδενική), εκτελούνται οι  
εντολές, αλλιώς  
τελειώνει η εκτέλεση της  
**for**



# Επαναληπτικές εντολές: for

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές

## Γενική μορφή

```
for (expression1; expression2; expression3;  
    statements;  
    )
```

Στο τέλος των εντολών **statements**, εκτελείται η έκφραση **expression3**



# Επαναληπτικές εντολές: for

- Εντολή **for**: επαναλαμβάνει κάποιες εντολές N φορές.

## Γενική μορφή

```
for (expression1; expression2; expression3 ) {  
    statements;  
}
```

## Ισοδύναμο

```
expression1;  
while(expression2) {  
    statements;  
    expression3;  
}
```





# Παράδειγμα 1: for

power.c

```
/* Συνάρτηση power
 * Υπολογίζει τη  $n$ -οστή δύναμη ενός αριθμού
 * Δέχεται τη βάση και τον εκθέτη
 * Επιστρέφει τη βάση υψωμένη στον εκθέτη
 */
int power(int base, int n)
{
    int i, result = 1;

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

    return result;
}
```



# Παράδειγμα 2: for

## factorial.c

```
/* Συνάρτηση factorial
 * Υπολογίζει το παραγοντικό ενός ακέραιου αριθμού
 * Δέχεται έναν ακέραιο αριθμό n
 * Επιστρέφει το παραγοντικό n!
 */
int factorial(int n)
{
    int i, result = 1;

    for(i = 2; i <= n; i++) {
        result = result * i;
    }

    return result;
}
```



# Παράδειγμα: `for` vs. `while`

## Παράδειγμα

```
int c;
for(c = 1; c <= 100; c++) {
    printf("The square of %d is %d\n", c, c * c);
}
```

## Παράδειγμα

```
int c;
c = 1;
while(c <= 10) {
    printf("The square of %d is %d\n", c, c * c);
    c++;
}
```

- Τα δύο παραδείγματα είναι ισοδύναμα
- Συνήθως οι προγραμματιστές προτιμούν το πρώτο
  - Σε μία γραμμή `for` υπάρχουν όλες οι πληροφορίες που περιγράφουν την επανάληψη



# Ειδικές περιπτώσεις for

- Στη **for** δεν είναι υποχρεωτικό να δωθούν οι **expression1**, **expression2** και **expression3**
- Τα παρακάτω επιτρέπονται στη C

## Ασυνήθιστο

```
int k;  
k = 40;  
for (;k;) {  
    printf("k = %d\n", k--);  
}
```

## Άδειο

```
for(;;);
```

- Τι υπολογίζουν τα παραπάνω παραδείγματα;



# Επαναληπτικές εντολές: `do while`

- Εντολή `do while`: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
do {  
    statements  
} while (expression);
```



# Επαναληπτικές εντολές: do while

- Εντολή **do while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

Γενική μορφή

```
do {  
    statements  
} while (expression);
```

Εκτελούνται οι εντολές  
(statements)



# Επαναληπτικές εντολές: do while

- Εντολή **do while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
do {  
    statements  
} while (expression);
```

Υπολογίζεται η τιμή της έκφρασης (expression) ως ακέραιος αριθμός, και συγκρίνεται με το 0



# Επαναληπτικές εντολές: do while

- Εντολή **do while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
do {  
    statements  
}  
while (expression);
```

Αν η έκφραση έχει τιμή  
διαφορετική από 0,  
εκτελούνται ξανά οι εντολές





# Επαναληπτικές εντολές: do while

- Εντολή **do while**: επαναλαμβάνει κάποιες εντολές για όσο ισχύει μια συνθήκη

## Γενική μορφή

```
do {  
    statements  
} while (expression);
```

Υπολογίζεται ξανά η τιμή της έκφρασης (expression) ως ακέραιος αριθμός, και συγκρίνεται με το 0, κλπ



# Παράδειγμα: do while

## dowhile.c

```
/* Συνάρτηση calculator: Δέχεται δύο ακέραιους αριθμούς, παίρνει από το χρήστη
 * το είδος της πράξης (+, -, * ή /) την εκτελεί και επιστρέφει το αποτέλεσμα.
 */
int calculator(int a, int b) {
    int result = 0;
    char choice;

    printf("Select + for addition, - for subtraction, * for multiplication or / for division\n");
    do {
        printf("Enter symbol:\n");
        choice = getchar();
    } while ((choice != '+') && (choice != '-') && (choice != '*') && (choice != '/'));
    if (choice == '+') {
        result = a + b;
    } else if (choice == '-') {
        result = a - b;
    } else if (choice == '*') {
        result = a * b;
    } else if (choice == '/') {
        result = a / b;
    }
    return result;
}
```

# Εντολές εξόδου από επανάληψη

## • Εντολή `break`

- Τερματίζει την εντολή επανάληψης (`for`, `while`, `do while`) στην οποία βρίσκεται
- Η εκτέλεση συνεχίζει με την επόμενη εντολή μετά το τέλος του `for`, `while` ή `do while`
- Αν υπάρχουν πάνω από μια *φωλιασμένες* εντολές επανάληψης, ισχύει για την πιο “μέσα”
- Χρησιμοποιείται για να τερματίσει την εντολή επανάληψης
  - Π.χ., βγές από το μενού, σταμάτα την αναζήτηση μόλις το βρείς, κλπ.

## • Εντολή `continue`

- Τερματίζει την συγκεκριμένη επανάληψη και συνεχίζει με την επόμενη
- Στις εντολές `for`, εκτελεί την `expression3` σαν να είχε φτάσει στο τέλος των `statements`
- Στις `while` και `do while` υπολογίζει τη συνθήκη επανάληψης για την επόμενη επανάληψη
- Χρησιμοποιείται για να παρακάμψει κάποια από όλες τις επαναλήψεις



# Φωλιασμένες εντολές επανάληψης

- Μπορεί μια εντολή επανάληψης να περιέχει άλλες εντολές επανάληψης

## Παράδειγμα

```
for (i = 1; i <= 8; i++) {  
    for (j = 1; j <= i; j++) {  
        printf("*");  
    }  
    printf("\n");  
}
```

## Έξοδος

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



# Παράδειγμα: Φωλιασμένοι βρόχοι

- Πώς μπορούμε να τυπώσουμε την ακόλουθη έξοδο;

## Ζητούμενη έξοδος

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

