

Προγραμματισμός Η/Υ Ι

# 05. Υπολογισμός - Εντολές Επανάληψης

ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2019-2020 | ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ Τ.Ε.

# Περίληψη

Σήμερα ...

- ▶ Θα συνεχίσουμε τη συζήτησή μας για τα **βασικά** στοιχεία του υπολογισμού
- ▶ Θα μελετήσουμε τον τρόπο **επανάληψης** εντολών σε ένα σύνολο τιμών (**iteration**)

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη

μερικές φορές πρέπει να **επαναλάβουμε** την εκτέλεση κάποιας (σύνθετης) εντολής

- ▶ είτε για ένα συγκεκριμένο **αριθμό** επαναλήψεων

- ▶ είτε για **όσο** ισχύει μια συνθήκη

- ❖ στη C αυτό μπορεί να γίνει με χρήση μίας εκ των εντολών:

- ▶ **for**

- ▶ **while**

- ▶ **do...while**

- ✍ ο **έλεγχος** της επανάληψης γίνεται με τη βοήθεια **λογικών εκφράσεων/προτάσεων**

- ✍ οι δομές που σχηματίζουν οι εντολές επανάληψης ονομάζονται «**βρόχοι**»

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - while

- ❖ το πρώτο πρόγραμμα που έτρεξε σε έναν υπολογιστή (David Wheeler, University of Cambridge, 1949)

```
1. // υπολογισμός και εκτύπωση ενός πίνακα τετραγώνων για τους αριθμούς 0-99
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i = 0; // αρχική τιμή μεταβλητής βρόχου, η οποία
8.               // παρακολουθεί πόσες φορές έχει εκτελεστεί ο βρόχος
9.     while (i < 100) { // συνθήκη τερματισμού βρόχου
10.         printf("%d \t %d\n", i, i*i); // εντολή βρόχου
11.         ++i; // αύξηση μεταβλητής βρόχου κατά 1
12.     }
13. }
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - while II

❖ Τι χρειαζόμαστε:

- ▶ μία μεταβλητή βρόχου (ή μεταβλητή ελέγχου)
- ▶ αρχικοποίηση μεταβλητής βρόχου
- ▶ κριτήριο τερματισμού
- ▶ αύξηση μεταβλητής βρόχου
- ▶ κάτι να κάνουμε σε κάθε επανάληψη

`i`

`int i = 0;`

`if i < 100 is false, terminate`

`++i;`

`printf(...);`

```
1.  int i = 0;
2.  while (i < 100) {
3.      printf("%d \t %d\n", i, i*i);
4.      ++i;
5.  }
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - Συντακτικό

❖ συντακτικό εντολής **while**:

**while** (condition)

statement

// while the condition is true, do statement

```
1. int i = 0;
2. while (i < 100) {
3.     printf("%d \t %d\n", i, i*i);
4.     ++i;
5. }
```

▶ η εντολή (statement) μπορεί να είναι απλή (μία) ή σύνθετη (μπλοκ)

▶ η εντολή εκτελείται όσο ισχύει η συνθήκη (condition)

✍ η συνθήκη αποτιμάται σε κάποια τιμή **a** και είναι

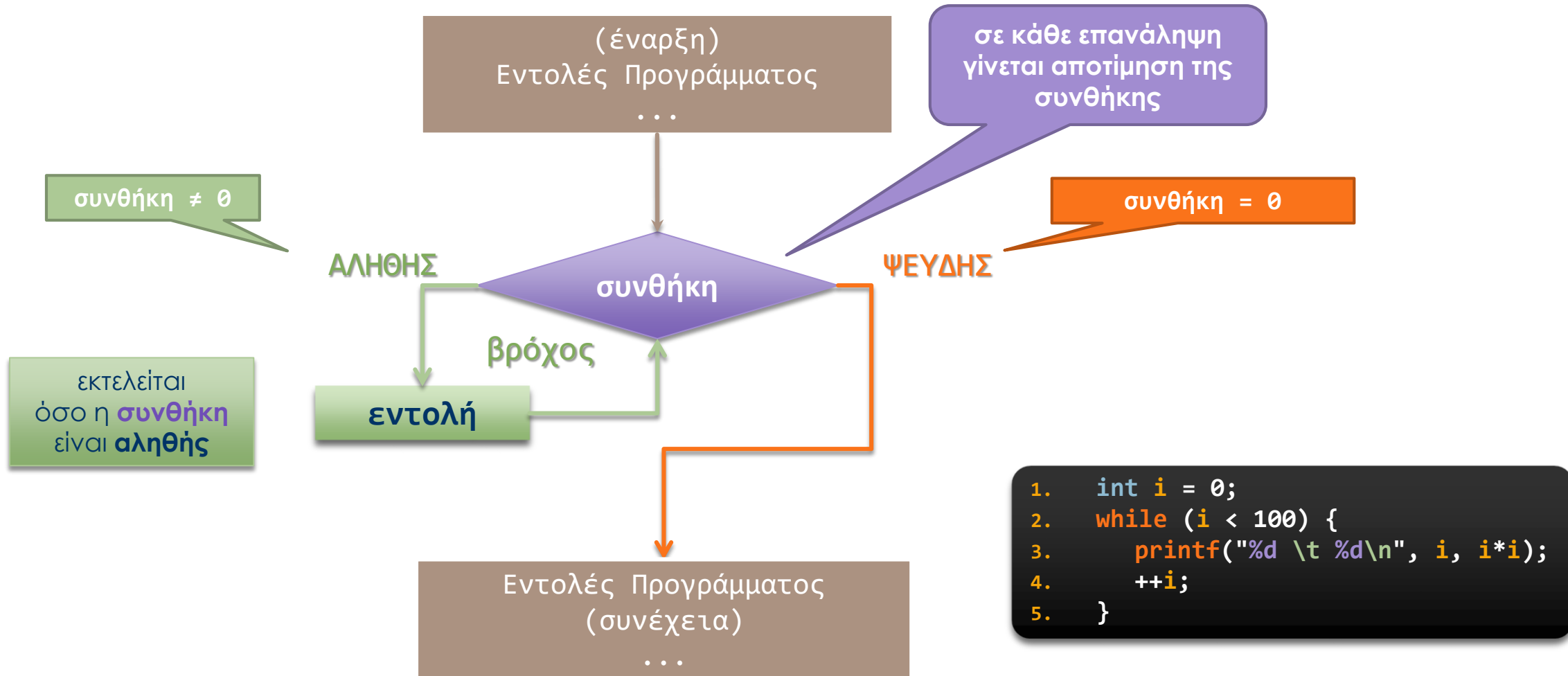
▶ αληθής όταν **a ≠ 0**

▶ ψευδής όταν **a = 0**

▶ όταν η συνθήκη γίνει ψευδής → σταματά η εκτέλεση της εντολής

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - Διάγραμμα ροής



# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - while - Παρατηρήσεις

❖ η **συνθήκη** μπορεί να είναι

1. μια έκφραση συσχετισμού

`x > 10`

2. μια λογική πρόταση

`(x > 10 && x < 20)`

3. το αποτέλεσμα κάποιας πράξης

`x + y`

4. μεταβλητή ή τιμή

`x` ή `100`

❖ αν η εντολή **δεν** είναι σύνθετη → τα άγκιστρα `{ }` **δεν** είναι απαραίτητα

❖ αν η **συνθήκη** είναι **ψευδής** από την **αρχή** → **δε** θα εκτελεστεί ποτέ η εντολή

```
while (2 > 3) statement
```

❖ αν η **συνθήκη** είναι πάντα **αληθής** → **ατέρμων** βρόχος

```
while (1) statement
```

❖ αν βάλουμε `';` μετά τη **συνθήκη** της **while** → η εντολή της **while** είναι η κενή εντολή!

```
while (x > 0);
```

`statement` // το statement είναι εκτός της while!

```
while (x > 0);  
statement
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - 1<sup>ο</sup> παράδειγμα

```
1. #include <stdio.h>
2.
3. main ()
4. {
5.     int i = 1;
6.     while (i < 5) {
7.         printf("%d\n", i);
8.         ++i;
9.     }
10.    printf("end\n");
11. }
```

i	(i<5)	printf
1	1	1
2	1	2
3	1	3
4	1	4
5	0	end

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - 2<sup>ο</sup> παράδειγμα

```
1. #include <stdio.h>
2.
3. main ()
4. {
5.     int i = 1;
6.     while (++i < 5) {
7.         printf("%d\n", i);
8.     }
9.     printf("end\n");
10. }
```

i	(++i<5)	printf
1	(2<5) → 1	2
2	(3<5) → 1	3
3	(4<5) → 1	4
4	(5<5) → 0	end

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - 3<sup>ο</sup> παράδειγμα

```
1. #include <stdio.h>
2.
3. main ()
4. {
5.     int i = 1;
6.     while (i++ < 5) {
7.         printf("%d\n", i);
8.     }
9.     printf("end\n");
10. }
```

i	(i++<5)	printf
1	(1<5) → 1	2
2	(2<5) → 1	3
3	(3<5) → 1	4
4	(4<5) → 1	5
5	(5<5) → 0	end

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - while - 4<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους που είναι μικρότεροι από το **n**

```
1. // εμφάνιση αριθμών μικρότερων του n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    i = 0;
11.    while (i < n) {
12.        printf("i is now: %d\n", i);
13.        ++i;
14.    }
15. }
```

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους από το **0** έως και το **n**

```
1. // εμφάνιση αριθμών από το 0 έως και το n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    i = 0;
11.    while (i <= n) {
12.        printf("i is now: %d\n", i);
13.        ++i;
14.    }
15. }
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - while - 5<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και υπολογίζει το άθροισμα: **1 + 2 + ... + n**

```
1.  int i, n, sum;
2.
3.  printf("Enter integer: ");
4.  scanf("%d", &n);
5.
6.  i = 0;
7.  sum = 0;
8.  while (i <= n) {
9.      sum += i;
10.     ++i;
11. }
12.
13. printf("Sum = %d\n", sum);
```

# Εντολή Επανάληψης

do..while

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - do..while - Συντακτικό

❖ ΣΥΝΤΑΚΤΙΚΟ εντολής **do...while**:

do

    statement           // do statement, while condition is true

while (condition);

- ▶ ο έλεγχος της συνθήκης (condition) γίνεται στο τέλος του βρόχου
- ▶ η εντολή (statement) εκτελείται τουλάχιστον μία φορά
- ▶ το **while** στο τέλος της εντολής **do...while** πρέπει να τελειώνει με ελληνικό ερωτηματικό (;)

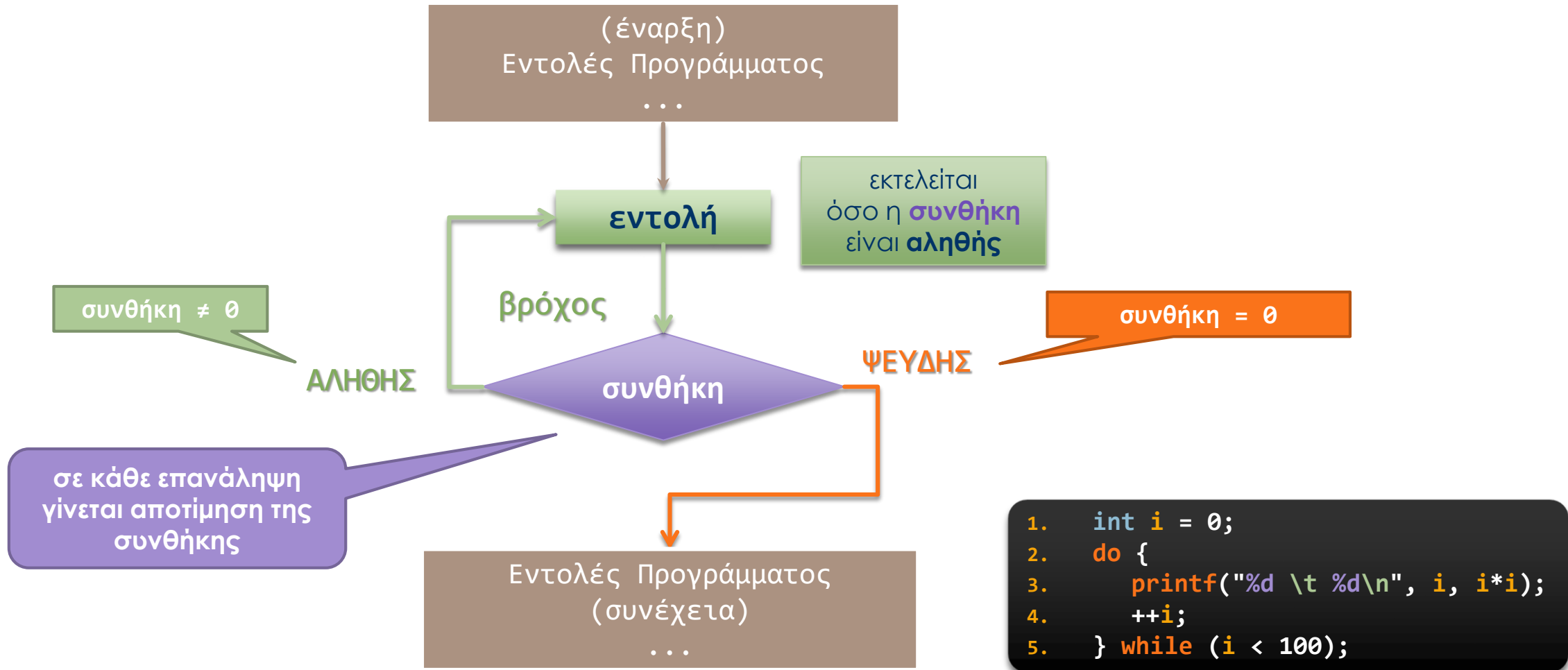
```
1. int i = 0;
2. while (i < 100) {
3.     printf("%d \t %d\n", i, i*i);
4.     ++i;
5. }
```



```
1. int i = 0;
2. do {
3.     printf("%d \t %d\n", i, i*i);
4.     ++i;
5. } while (i < 100);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - do..while - Διάγραμμα ροής



# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - do..while - 1<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους που είναι μικρότεροι από το **n**

```
1. // εμφάνιση αριθμών μικρότερων του n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    i = 0;
11.    do {
12.        printf("i is now: %d\n", i);
13.        ++i;
14.    } while (i < n);
15. }
```

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους από το **0** έως και το **n**

```
1. // εμφάνιση αριθμών από το 0 έως και το n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    i = 0;
11.    do {
12.        printf("i is now: %d\n", i);
13.        ++i;
14.    } while (i <= n);
15. }
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - do..while - 2<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και υπολογίζει το άθροισμα:  $1 + 2 + \dots + n$

```
1.  int i, n, sum;
2.
3.  printf("Enter integer: ");
4.  scanf("%d", &n);
5.
6.  i = 0;
7.  sum = 0;
8.  do {
9.      sum += i;
10.     ++i;
11. } while (i <= n);
12.
13. printf("Sum = %d\n", sum);
```

# Εντολή Επανάληψης

for

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for

- ❖ η εντολή **for** είναι παρόμοια με την εντολή **while**, με τη διαφορά ότι η διαχείριση της μεταβλητής ελέγχου επικεντρώνεται στην **αρχή** → **ευκολότερο** να καταλάβουμε τι συμβαίνει

```
1. for (int i = 0; i < 100; ++i)
2.     printf("%d \t %d\n", i, i*i);
```



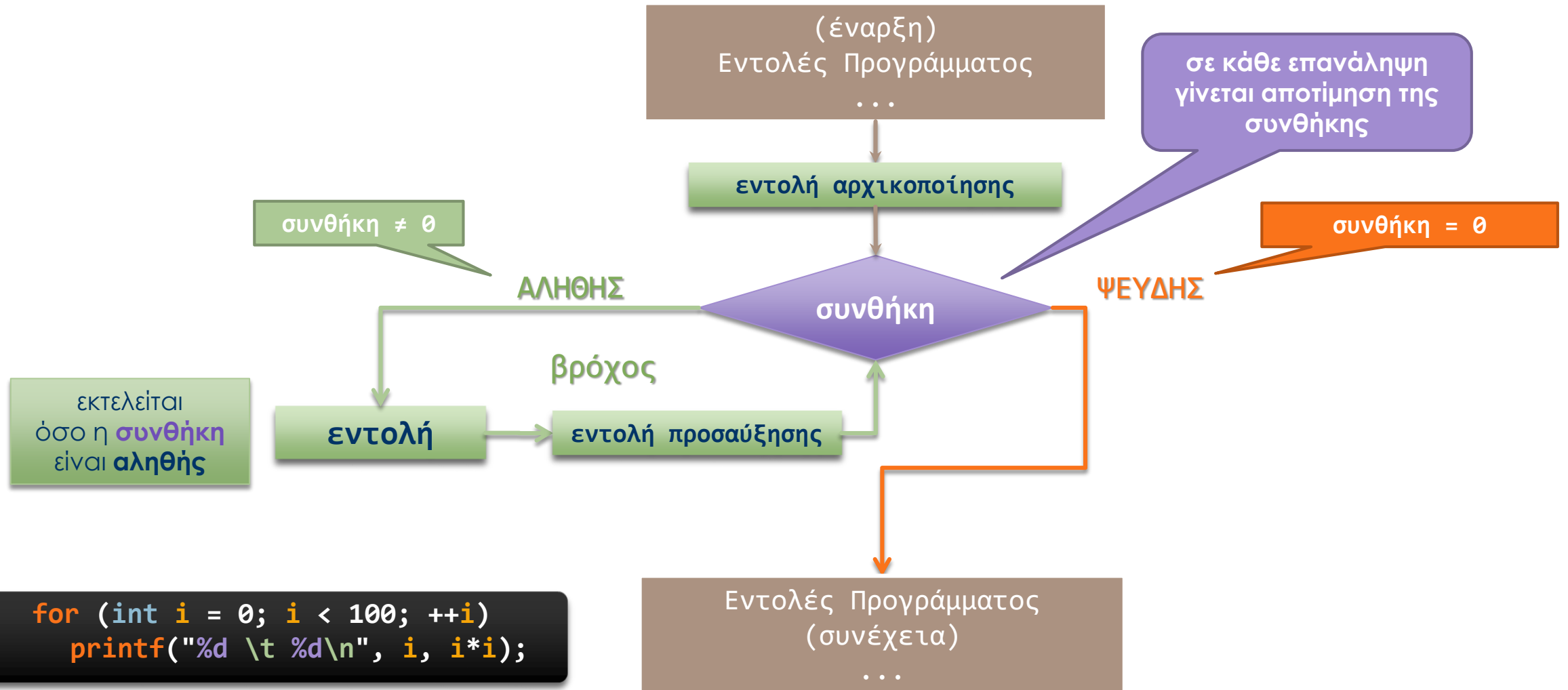
```
1. int i = 0;
2. while (i < 100) {
3.     printf("%d \t %d\n", i, i*i);
4.     ++i;
5. }
```

- ❖ **ΣΥΝΤΑΚΤΙΚΟ** εντολής **for**:

```
for (initialize; condition; increment)
    statement
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - for - Διάγραμμα ροής



1. `for (int i = 0; i < 100; ++i)`
2. `printf("%d \t %d\n", i, i*i);`

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for - Παρατηρήσεις

❖ **ΣΥΝΤΑΚΤΙΚΟ** εντολής **for**:

```
for (initialize; condition; increment)  
    statement
```

- ▶ αν η **συνθήκη** είναι **ψευδής** δε θα εκτελεστεί ποτέ η **εντολή**
- ▶ η εντολή **αρχικοποίησης** (initialize) μπορεί να είναι **περισσότερες** από μία εντολές

```
for (i = 0, j = 0, k = 1; i < k; ++i) {...}
```

- ▶ η εντολή **προσαύξησης** (increment) μπορεί να είναι **περισσότερες** από μία εντολές

```
for (i = 0; i < k; ++i, j+=2) {...}
```

- ✍️ όταν οι εντολές αρχικοποίησης και προσαύξησης αποτελούνται από **πολλαπλές** εντολές τότε αυτές **χωρίζονται** μεταξύ τους με **κόμμα** (,)

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for - Παρατηρήσεις II

- ▶ χρησιμοποιούμε τη **for** όταν γνωρίζουμε εκ των προτέρων τον αριθμό επαναλήψεων

- ✖ δε βάζουμε το ελληνικό ερωτηματικό (;) στο τέλος της **for**

```
for (i = 0; i < 100; ++i); {...} // απλά αυξάνει το i 100 φορές
```

- ▶ δεν είναι απαραίτητο να υπάρχουν και τα τρία τμήματα της **for**

- ▶ όμως υπάρχει πάντα το διαχωριστικό ';' μεταξύ των τμημάτων

```
for (;;) {...} // ατέρμων βρόχος
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for

? παρατηρείτε κάποιο πρόβλημα στον παρακάτω κώδικα;

```
1.   for (int i = 0; i < 100; ++i)    // για κάθε i στην περιοχή [0:100)
2.   {
3.       printf("%d \t %d\n", i, i*i);
4.       ++i;
5.   }
```

❖ δεν τροποποιούμε **ΠΟΤΕ** το μετρητή του βρόχου στο μπλοκ εντολών της **for** → **δυσνόητος** κώδικας

▶ εάν θέλετε να αυξήσετε το μετρητή κατά 2, **ΠΕΙΤΕ ΤΟ!**

```
1.   // υπολογισμός και εκτύπωση ενός πίνακα τετραγώνων άρτιων αριθμών στην περιοχή [0:100)
2.
3.   for (int i = 0; i < 100; i+=2)
4.       printf("%d \t %d\n", i, i*i);
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for - 1<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους που είναι μικρότεροι από το **n**

```
1. // εμφάνιση αριθμών μικρότερων του n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    for (i = 0; i < n; ++i)
11.        printf("i is now: %d\n", i);
12. }
```

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και να εμφανίζει όλους τους ακεραίους από το **0** έως και το **n**

```
1. // εμφάνιση αριθμών από το 0 έως και το n
2.
3. #include <stdio.h>
4.
5. main ()
6. {
7.     int i, n;
8.     printf("Enter integer: ");
9.     scanf("%d", &n);
10.    for (i = 0; i <= n; ++i)
11.        printf("i is now: %d\n", i);
12. }
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - for - 2<sup>ο</sup> παράδειγμα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη έναν ακέραιο **n** και υπολογίζει το άθροισμα: **1 + 2 + ... + n**

```
1.  int i, n, sum;
2.
3.  printf("Enter integer: ");
4.  scanf("%d", &n);
5.
6.  for (i = 0, sum = 0; i <= n; ++i)
7.      sum += i;
8.
9.  printf("Sum = %d\n", sum);
```

# while, do..while, for

Παραδείγματα

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Παρουσίαση αριθμών από 0 έως n

```
1. int i, n;  
2. printf("Enter integer: ");  
3. scanf("%d", &n);  
4. i = 0;  
5. while (i <= n) {  
6.     printf("i is now: %d\n", i);  
7.     ++i;  
8. }
```

```
1. int i, n;  
2. printf("Enter integer: ");  
3. scanf("%d", &n);  
4. i = 0;  
5. do {  
6.     printf("i is now: %d\n", i);  
7.     ++i;  
8. } while (i <= n);
```

```
1. int i, n;  
2. printf("Enter integer: ");  
3. scanf("%d", &n);  
4. for (i = 0; i <= n; ++i)  
5.     printf("i is now: %d\n", i);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Άθροισμα αριθμών  $1+2+\dots+n$

```
1. int i, n, sum;
2. printf("Enter integer: ");
3. scanf("%d", &n);
4. i = 0;
5. sum = 0;
6. while (i <= n) {
7.     sum += i;
8.     ++i;
9. }
10. printf("Sum = %d\n", sum);
```

```
1. int i, n, sum;
2. printf("Enter integer: ");
3. scanf("%d", &n);
4. i = 0;
5. sum = 0;
6. do {
7.     sum += i;
8.     ++i;
9. } while (i <= n);
10. printf("Sum = %d\n", sum);
```

```
1. int i, n, sum;
2. printf("Enter integer: ");
3. scanf("%d", &n);
4. for (i = 0, sum = 0; i <= n; ++i)
5.     sum += i;
6. printf("Sum = %d\n", sum);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Έυρεση αθροίσματος N ακεραίων (for)

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη N ακεραίους και να υπολογίζει το άθροισμά τους

```
1.  int i, n, N, sum;
2.
3.  printf("Enter number of integers (N): ");
4.  scanf("%d", &N);
5.
6.  for (i = 0, sum = 0; i < N; ++i) {
7.      printf("Enter integer: ");
8.      scanf("%d", &n);
9.      sum += n;
10. }
11.
12. printf("Sum = %d\n", sum);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Έυρεση αθροίσματος N πραγματικών (while)

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη N πραγματικούς αριθμούς και να υπολογίζει το άθροισμά τους

```
1.  int i, N;
2.  float n, sum;
3.
4.  printf("Enter number of integers (N): ");
5.  scanf("%d", &N);
6.
7.  i = 0;
8.  sum = 0;
9.  while (i < N) {
10.     printf("Enter real: ");
11.     scanf("%f", &n);
12.     sum += n;
13.     ++i;
14. }
15.
16. printf("Sum = %f\n", sum);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Έυρεση μέσου όρου N ακεραίων (do..while)

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη N ακεραίους αριθμούς και να υπολογίζει το μέσο όρο τους

```
1. int i, n, N, sum;
2. float average;
3.
4. printf("Enter number of integers (N): ");
5. scanf("%d", &N);
6.
7. i = 0;
8. sum = 0;
9. do {
10.     printf("Enter integer: ");
11.     scanf("%d", &n);
12.     sum = sum + n;
13.     ++i;
14. } while (i < N);
15.
16. average = (float)sum/N;
17.
18. printf("Average = %f\n", average);
```

ο μέσος όρος (ως αποτέλεσμα διαίρεσης) είναι πραγματικός αριθμός

μετατροπή της τιμής της sum σε πραγματικό αριθμό → ώστε να έχουμε πραγματικό αριθμό ως αποτέλεσμα

break & continue

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - break

- ❖ έχει ως αποτέλεσμα την έξοδο του προγράμματος από τον (πιο εσωτερικό) βρόχο ή `switch` στο οποίο αυτή περιέχεται
  - ▶ το πρόγραμμα **συνεχίζει** με τις εντολές που ακολουθούν μετά το βρόχο

```
1.  int i, N, sum;
2.
3.  printf("Enter integer (N): ");
4.  scanf("%d", &N);
5.
6.  for (i = 0, sum = 0; i < N; ++i) {
7.      if (i == 5) break;
8.      sum = sum + i;
9.  }
10.
11. printf("Sum = %d\n", sum);
```

Θα υπολογίσει το πολύ το άθροισμα των αριθμών 1+2+3+4, ανεξάρτητα από την τιμή του N (καθώς ο βρόχος σταματά όταν το i πάρει την τιμή 5)

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - continue

- ❖ επιστρέφει τη ροή του προγράμματος στον έλεγχο της συνθήκης σε ένα βρόχο (τον πιο εσωτερικό)
  - ▶ δε διακόπτεται ο βρόχος, αλλά η τρέχουσα επανάληψη

```
1.  int i, N, sum;
2.
3.  printf("Enter integer (N): ");
4.  scanf("%d", &N);
5.
6.  for (i = 0, sum = 0; i < N; ++i) {
7.      if (i == 5) continue;
8.      sum = sum + i;
9.  }
10.
11. printf("Sum = %d\n", sum);
```

Θα υπολογίσει το άρθιοισμα:  $1+2+\dots+n-5$   
(όταν το  $i$  πάρει την τιμή 5, η γραμμή 8 δε θα εκτελεστεί, όμως ο βρόχος δε θα τερματιστεί, αλλά θα συνεχίσει για  $i = 6, \dots$ )

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - break & continue

### break

- ▶ έχει ως αποτέλεσμα την **έξοδο** του προγράμματος **από τον** (πιο εσωτερικό) βρόχο ή **switch** στο οποίο αυτή περιέχεται
  - ▶ το πρόγραμμα **συνεχίζει** με τις εντολές που ακολουθούν μετά το βρόχο

### continue

- ❖ **επιστρέφει** τη ροή του προγράμματος στον **έλεγχο** της **συνθήκης** σε ένα βρόχο (τον πιο εσωτερικό)
- ▶ βολικό για
  1. τον έλεγχο **συνθηκών πριν** ξεκινήσει η εκτέλεση του βρόχου, ή
  2. για πρόωρη επιστροφή στον έλεγχο της **συνθήκης**

✍ βρόχοι: **for, while, do ... while**

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - break - 1<sup>ο</sup> Παράδειγμα

```
while (... && !StopFlag)
{
    <some code>

    if (I should stop){
        StopFlag = true;
    }else{
        <some more code>
    }
} // end of while loop
```



```
while (...)
{
    <some code>

    if (I should stop){
        break;
    }

    < some more code>
} // end of while loop
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - continue - Παράδειγμα

```
while (...)  
{  
    if (everything is ok){  
        <rest of code>  
    } // end of if  
} // end of while loop
```



```
while (...)  
{  
    if (I don't like something){  
        continue;  
    }  
    <rest of code>  
} // end of while loop
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - break - 2<sup>ο</sup> Παράδειγμα

```
1. // Παράδειγμα χρήσης break
2.
3. main()
4. {
5.     int counter = 0;
6.
7.     while (1) {                // ατέρμων βρόχος
8.         counter++;
9.         if (counter == 100)    // αύξηση counter έως το 100
10.            break;            // έξοδος από τον ατέρμων βρόχο
11.     }
12.
13.     printf("the value of counter is: %d\n", counter);
14. }
```

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - break - 3<sup>ο</sup> Παράδειγμα

```
1. // Παράδειγμα χρήσης break
2.
3. main()
4. {
5.     int counter1 = 0, counter2 = 0;
6.
7.     while (1) { // 1ος ατέρμων βρόχος
8.         counter1++;
9.         if (counter1 == 100) // αύξηση counter1 έως το 100
10.            break; // έξοδος από τον 1ο ατέρμων βρόχο
11.
12.        counter2 = 0;
13.        while (1) { // 2ος ατέρμων βρόχος
14.            counter2+=2;
15.            if (counter2 == 200) // αύξηση counter2 έως το 200
16.                break; // έξοδος από τον 2ο ατέρμων βρόχο
17.        }
18.    }
19. }
```

# while, do..while, for

Περισσότερα παραδείγματα

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Έυρεση πλήθους αριθμών με κάποια ιδιότητα

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη **N** ακεραίους και να υπολογίζει το πλήθος των αριθμών που είναι μεγαλύτεροι του 100

```
1.  int i, N, n, counter;
2.
3.  printf("Enter number of integers (N): ");
4.  scanf("%d", &N);
5.
6.  i = 0;
7.  counter = 0;
8.  while (i < N) {
9.      printf("Enter integer: ");
10.     scanf("%d", &n);
11.     if (n > 100)
12.         ++counter;
13.     ++i;
14. }
15.
16. printf("Read %d numbers larger than 100\n", counter);
```

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Έυρεση μικρότερου αριθμού

- ▶ γράψτε ένα πρόγραμμα το οποίο να ζητά από το χρήστη **N** ακεραίους και να εκτυπώνει τον μικρότερο

```
1.  int i, N, n, min;
2.
3.  printf("Enter number of integers (N): ");
4.  scanf("%d", &N);
5.
6.  for (i = 0; i < N; ++i) {
7.      printf("Enter integer: ");
8.      scanf("%f", &n);
9.      if (i == 0)
10.         min = n;
11.     else if (n < min)
12.         min = n;
13. }
14.
15. printf("The minimum number is: %d\n", min);
```

# Ένθετοι βρόχοι

# Χαρακτηριστικά γλώσσας

## Εντολή (statement) - Επανάληψη - Ένθετοι βρόχοι

ένας βρόχος μπορεί να περιέχει άλλες εντολές βρόχων

- ▶ βρόχοι μέσα σε βρόχο → **ένθετοι** βρόχοι
- ▶ σε αυτές τις περιπτώσεις θα πρέπει να είμαστε **προσεκτικοί** στους **μετρητές** που χρησιμοποιούνται στους βρόχους

# Χαρακτηριστικά γλώσσας

Εντολή (statement) - Επανάληψη - Ένθετοι βρόχοι - 1<sup>ο</sup> Παράδειγμα

- ▶ γράψτε ένα πρόγραμμα που εκτυπώνει την προπαίδεια 1-10

```
1. main()
2. {
3.     int i, j;
4.
5.     for (i = 1; i <= 10; ++i) {
6.         for (j = 1; j <= 10; ++j)
7.             printf("%d\t", i*j);
8.         printf("\n");
9.     }
10. }
```

- ? τι θα άλλαζε για την προπαίδεια 1-5;

```
5.     for (i = 1; i <= 5; ++i) {
6.         for (j = 1; j <= 10; ++j)
7.             printf("%d\t", i*j);
8.         printf("\n");
9.     }
```

# Σύνοψη

- ▶ Εντολές επανάληψης
  - ▶ while
  - ▶ do..while
  - ▶ for
- ▶ Διαγράμματα ροής
- ▶ Ένθετοι βρόχοι
- ▶ break, continue