Η γλώσσα προγραμματισμού C

Οι εντολές επανάληψης
(while, do-while, for)

Κ. Βασιλάκης, ΣΤΕΦ, ΤΕΙ Κρήτης

Γενικά για τις εντολές επανάληψης

- Συχνά στο προγραμματισμό είναι επιθυμητή η πολλαπλή εκτέλεση μιας ενότητας εντολών, είτε
  - για ένα συγκεκριμένο αριθμό επαναλήψεων, είτε
  - όσο ισχύει κάποια συνθήκη.
- Τέτοιου είδους εντολές στη C είναι:
  - **while**
  - **do - while**
  - **for**
  - **goto** (δεν την χρησιμοποιούμε…)
- Ο έλεγχος της επανάληψης γίνεται με τη βοήθεια λογικών εκφράσεων/προτάσεων.
- Οι δομές που σχηματίζουν οι εντολές επανάληψης ονομάζονται «βρόγχοι».

Η σύναξη της εντολής while

- Η γενική σύνταξη της εντολής είναι:

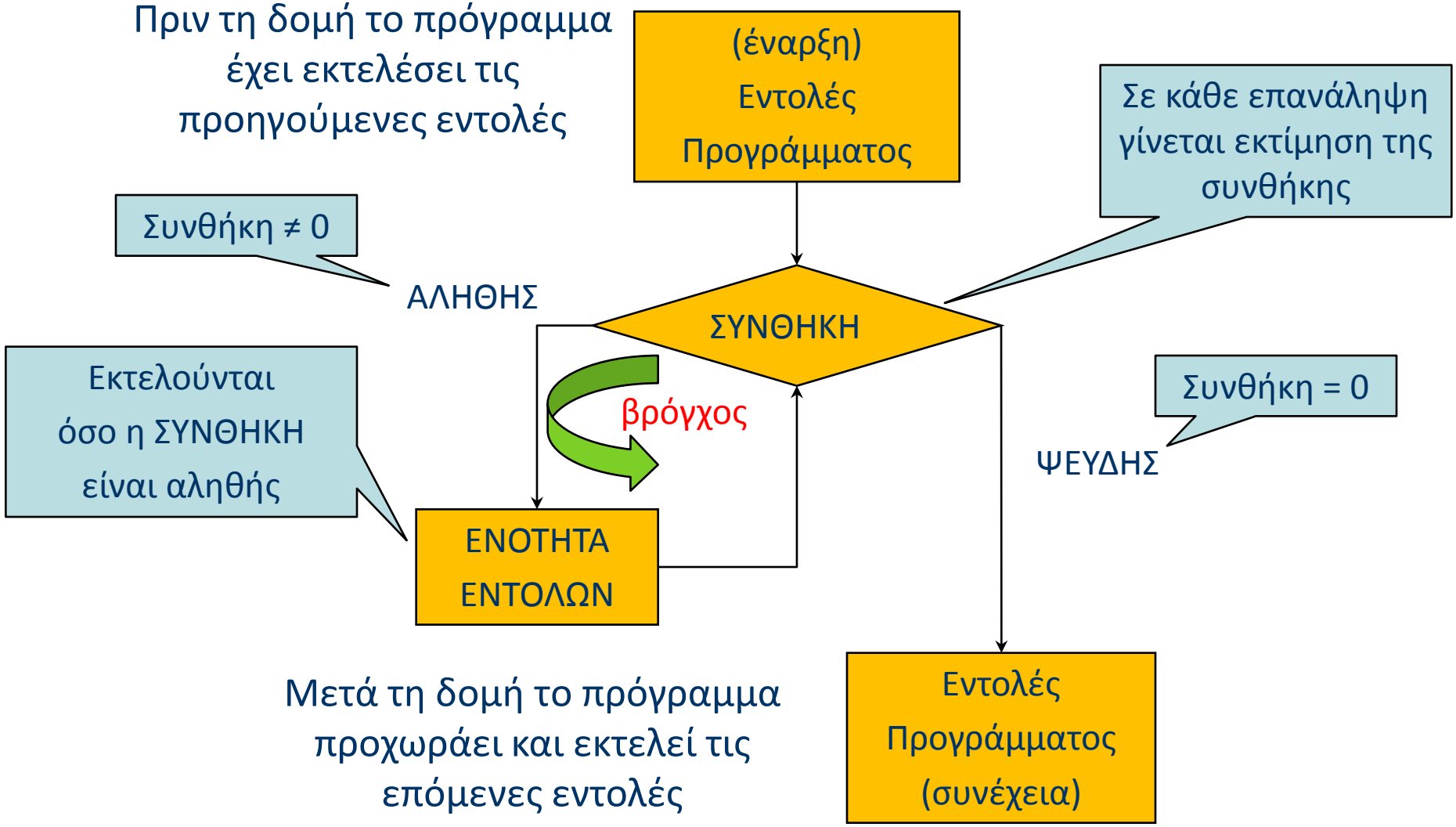
  while (<ΣΥΝΘΗΚΗ>)

  ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ

- Η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ*** μπορεί να περιλαμβάνει μία εντολή ή πολλές εντολές (block) που περικλείονται σε άγκιστρα ({,}).
- Οι εντολές (ή η εντολή) της ***ΕΝΟΤΗΤΑΣ_ΕΝΤΟΛΩΝ*** εκτελούνται όσο ισχύει η <***ΣΥΝΘΗΚΗ***> (είναι αληθής). Θυμηθείτε:
  - <ΣΥΝΘΗΚΗ> αληθής: διάφορη του μηδενός
  - <ΣΥΝΘΗΚΗ> ψευδής: ίση με μηδέν
- Όταν η <***ΣΥΝΘΗΚΗ***> γίνει ψευδής, τότε σταματά η εκτέλεση των εντολών της ενότητας.

Στη while η <***ΣΥΝΘΗΚΗ***> ελέγχεται στη αρχή του βρόγχου. Άρα αν η <***ΣΥΝΘΗΚΗ***> είναι ψευδής πριν ξεκινήσει ο βρόγχος, τότε οι εντολές της ***ΕΝΟΤΗΤΑΣ_ΕΝΤΟΛΩΝ*** δεν εκτελούνται ποτέ (ούτε μια φορά).

Γραφική αναπαράσταση της δομής while



Παρατηρήσεις στη while

- Η <***ΣΥΝΘΗΚΗ***> μπορεί να είναι λογική πρόταση, έκφραση συσχετισμού, αποτέλεσμα κάποιας πράξης, είτε ακόμα και κάποια μεταβλητή ή αριθμός.
- Αν η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ*** περιέχει μόνο μία εντολή τότε τα άγκιστρα ({,}) μπορούν να παραλειφθούν.
- Αν βάζουμε το ελληνικό ερωτηματικό «**;**» στο τέλος της ***while*** τότε η εντολή θεωρείται ξεχωριστή και ουσιαστικά δεν εκτελείται η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ***.
- Αν η <***ΣΥΝΘΗΚΗ***> σε μια ***while*** είναι πάντα αληθής, τότε ο βρόγχος δεν τερματίζει ποτέ (ατέρμων βρόγχος – infinity loop). Παράδειγμα ατέρμονος βρόγχου: ***while (1)***
- Αν η <***ΣΥΝΘΗΚΗ***> είναι ψευδής από την αρχή, τότε δεν θα εκτελεστεί ποτέ η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ***.

Παράδειγμα -1 (while)

Εκτέλεση

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

```
#include <stdio.h>
main()
{
    int i =1;
    while (i<5)
    {
        printf("%d\n",i);
        i=i+1;
    }
    printf("end");
getchar(); getchar();
}
```

Δοκιμάστε:

```
while (++i<5)
    printf("%d\n",i);
```

```
while (i++<5)
    printf("%d\n",i);
```

Παράδειγματα -2 (while)

Παρουσίαση αριθμών από 0 έως N. Το N δίνεται.

```
int i,N;
printf("Dwste N:");
scanf("%d", &N);
i = 0;
while (i < N) {
    printf("i is now %d!\n", i);
     i++;
}
```

Εύρεση αθροίσματος 1+2+3+...+N

```
int i,N, sum;
printf("Dwste N:");
scanf("%d", &N);
 i = 1;
 sum=0;
  while (i <= N) {
         sum=sum+i;
         i++;
  }
printf("SUM=%d\n", sum);
```

Η εντολή do -while

- Η γενική σύνταξη της εντολής είναι:

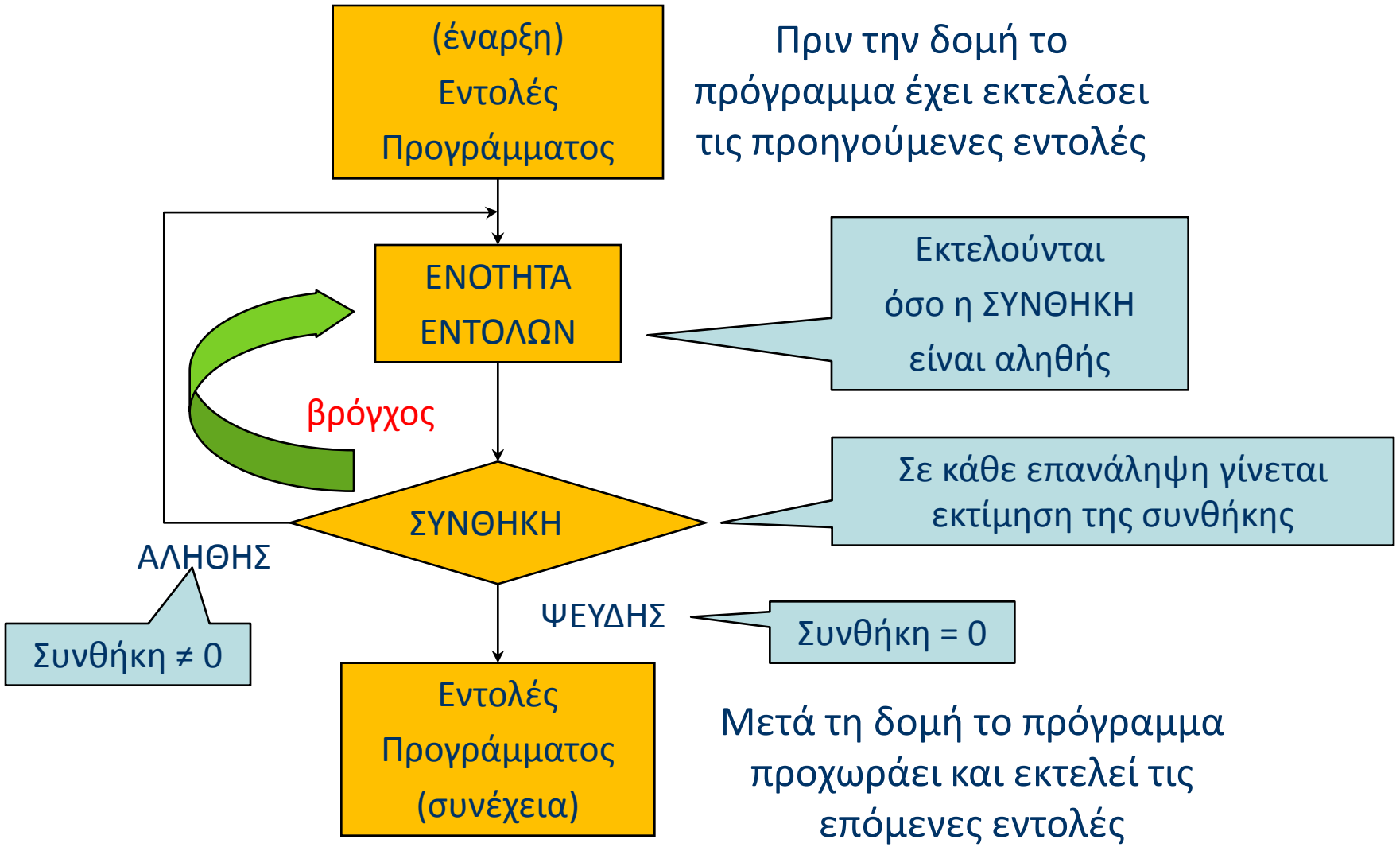
  do

  ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ

  while (<ΣΥΝΘΗΚΗ>);

- Συνεχής εκτέλεση της ***ΕΝΟΤΗΤΑΣ_ΕΝΤΟΛΩΝ*** μέχρι η <***ΣΥΝΘΗΚΗ***> να μην είναι αληθής.
- Ο έλεγχος εδώ γίνεται στο τέλος του βρόγχου (αντίθετα με τη while).
- Η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ*** εκτελείται τουλάχιστον μία φορά (ακόμα και αν η <***ΣΥΝΘΗΚΗ***> είναι ψευδής).
- Το ***while*** στο τέλος της εντολής ***do-while*** πρέπει να τελειώνει με το ελληνικό ερωτηματικό (**;**)

Γραφική αναπαράσταση της δομής do-while



Παράδειγματα (do - while)

Παρουσίαση αριθμών από 0 έως N. Το N δίνεται.

```
int i,N;
printf("Dwste N:");
scanf("%d", &N);
i = 0;
do {
    printf("i is now %d!\n", i);
     i++;
}
while (i<N);
```

Εύρεση αθροίσματος 1+2+3+…+N

```
int i,N, sum;
printf("Dwste N:");
scanf("%d", &N);
 i = 1;  sum=0;
  do {
        sum=sum+i;
        i++;
} while (i <= N);
printf("SUM=%d\n", sum);
```

Η εντολή for

- Γενική σύνταξη της for:

for (Εντολή1; <ΣΥΝΘΗΚΗ>; Εντολή2)

ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ

- Η ***Εντολή1*** μπορεί να είναι και περισσότερες από μία εντολές που συνήθως καθορίζουν κάποιες αρχικές τιμές. Εκτελείται μία φορά.
- Η ***Εντολή2*** είναι συνήθως μια παράσταση που συνήθως αλλάζει τη τιμή μιας μεταβλητής που βρίσκεται στη <***ΣΥΝΘΗΚΗ***>. Μπορεί να είναι και πάνω από μία εντολή.
- Η ανάπτυξη της εντολής ***for*** είναι η εξής:

Εκτέλεση της ***Εντολής1*** και στη συνέχεια εκτέλεση της ***ΕΝΟΤΗΤΑΣ_ΕΝΤΟΛΩΝ*** και της ***Εντολής2,*** όσο η **<ΣΥΝΘΗΚΗ>** είναι αληθής

Παρατηρήσεις για τη for

- Χρησιμοποιούμε την ***for*** όταν γνωρίζουμε εκ των προτέρων τον αριθμό επαναλήψεων.
- Δεν βάζουμε το ελληνικό ερωτηματικό «**;**» στο τέλος της ***for***:

  for (i = 0; i< 1000; i++); *// απλά αυξάνει το i 1000 φορές*

- Όταν η ***Εντολή1*** και η ***Εντολή2*** αποτελούνται από πολλές εντολές τότε αυτές χωρίζονται με κόμμα (**,**):

  for (i=0, k=0; (i < 5 && k < 3); i++, k++)

- Δεν είναι απαραίτητο να υπάρχουν και τα 3 τμήματα της ***for***. Όμως υπάρχει πάντα το διαχωριστικό «**;**» μεταξύ των τμημάτων.

  for (;;;) { εντολές } *// ατέρμων βρόγχος – infinity loop*

- Φυσικά αν η <***ΣΥΝΘΗΚΗ***> είναι ψευδής δεν θα εκτελεστεί ποτέ η ***ΕΝΟΤΗΤΑ_ΕΝΤΟΛΩΝ***.

Παραδείγματα με τη for

Παρουσίαση αριθμών από 0 έως N. Το N δίνεται.

```
int i,N;
printf("Dwste N:");
scanf("%d", &N);
for (i = 0; i<N;i++)
{
   printf("i is now %d!\n", i);
}
```

Εύρεση αθροίσματος 1+2+3+..+N

```
int i,N, sum;
printf("Dwste N:");
scanf("%d", &N);
sum=0;
 for (i = 0; i<=N;i++)
        sum=sum+i;
printf("SUM=%d\n", sum);
```

Παρουσίαση αριθμών από 0 έως N. Το N δίνεται.

```
printf("Dwste N:");
scanf("%d", &N);
i = 0;
while (i < N) {
    printf("i is now %d!\n", i);
    i++;
}
```

```
printf("Dwste N:");
scanf("%d", &N);
for (i = 0; i<N;i++)
{
    printf("i is now %d!\n", i);
}
```

```
printf("Dwste N:");
scanf("%d", &N);
i = 0;
do {
    printf("i is now %d!\n", i);
    i++;
} while (i<N);
```

Εύρεση αθροίσματος 1+2+3+...+N

```
printf("Dwste N:");
scanf("%d", &N);
 i = 1;  sum=0;
  do {
        sum=sum+i;
        i++;
}while (i <= N);
printf("SUM=%d\n", sum);
```

```
printf("Dwste N:");
scanf("%d", &N);
sum=0;
for (i = 0; i<=N;i++)
        sum=sum+i;
printf("SUM=%d\n", sum);
```

```
printf("Dwste N:");
scanf("%d", &N);
 i = 1;
 sum=0;
  while (i <= N) {
        sum=sum+i;
        i++;
  }
printf("SUM=%d\n", sum);
```

Εύρεση αθροίσματος Ν ακεραίων (for)

```
int i,N,A,sum;
printf("Dwste N:");
scanf("%d", &N);
sum=0;
for (i = 1; i<=N;i++)
{
        printf("Dwste akeraio:");
        scanf("%d",&A);
        sum=sum+A;
}
printf("SUM=%d\n", sum);
```

Εύρεση αθροίσματος Ν πραγματικών (while)

```
int i, N;
float F, sum;
printf("Dwste N:");  scanf("%d", &N);
i=1;
sum=0;
while  (i<=N) {
        printf("Dwste pragmatiko:");
        scanf("%f",&F);
        sum=sum+F;
         i++;
}
printf("SUM=%f\n", sum);
```

Εύρεση Μέσου Όρου Ν πραγματικών (do-while)

```
int i, N;
float F, sum, MO;
printf("Dwste N:");  scanf("%d", &N);
i=1;
sum=0;
do {
        printf("Dwste pragmatiko:");  scanf("%f",&F);
        sum=sum+F;
         i++;
} while  (i<=N) ;
MO=sum/N;
printf("Mesos Oros=%f\n", MO);
```

Ο μέσος όρος είναι πάντα πραγματικός (αποτέλεσμα διαίρεσης)

Η εντολή break

- Χρησιμοποιείται για ηθελημένο τερματισμό ενός βρόχου (for, while ή do-while). Επίσης για το τερματισμό της switch.
- Η εκτέλεση του break μέσα στο βρόγχο μεταφέρει τον έλεγχο του προγράμματος στην εντολή που ακολουθεί τον βρόγχο.

```
int i,N, sum;
printf("Dwste N:");  scanf("%d", &N);
sum=0;
 for (i = 0; i<=N;i++) {
          if (i==5) break;
         sum=sum+i;
 }
printf("SUM=%d\n", sum);
```

Θα βρει το πολύ το άθροισμα των 1+2+3+4, ανεξάρτητα από την τιμή του N

Η εντολή continue

- Στο σημείο που χρησιμοποιείται στους βρόγχους διακόπτεται η εκτέλεση της ***ΕΝΟΤΗΤΑΣ_ΕΝΤΟΛΩΝ*** και ξεκινά την επόμενη επανάληψη.
- Δεν διακόπτεται ο βρόγχος αλλά η τρέχουσα επανάληψη.

```
int i,N, sum;
printf("Dwste N:");  scanf("%d", &N);
sum=0;
 for (i = 0; i<=N;i++) {
          if (i>=5) continue;
         sum=sum+i;
 }
printf("SUM=%d\n", sum);
```

Θα βρει το πολύ το άθροισμα των 1+2+3+4, ανεξάρτητα από την τιμή του N

Να βρεθεί το πλήθος των αριθμών που χαρακτηρίζεται από κάποια ιδιότητα (πχ μεγαλύτεροι του 100)

```
int i,N,A,M;
printf("Dwste N:");
scanf("%d", &N);
i=1;
M=0;
while(i<=N)
{
        printf("Dwste akeraio:");
        scanf("%d",&A);
        if (A>100) M++;
         i++;
}
printf("Diavasa %d arithmous >100\n", M);
```

Χρησιμοποιούμε κάποια μεταβλητή σαν μετρητή

Ο μετρητής είναι πάντα ακέραιος και δεν ξεχνάμε να τον μηδενίζουμε πριν.

Αυξάνω τον μετρητή Μ κάθε φορά που βρίσκω την ιδιότητα

Διαβάστε Ν αριθμούς και βρείτε τον μικρότερο τους

```
int i,N,A,min;
printf("Dwste N:");
scanf("%d", &N);
printf("Dwste akeraio:");
scanf("%d",&A);
min=A;
for (i = 2; i<=N;i++)
{
        printf("Dwste akeraio:");
        scanf("%d",&A);
        if (A<min) min=A;
}
printf("To min einai: %d \n", min);
```

Διαβάζουμε ξεχωριστά τον 1ο αριθμό

Δίνουμε σαν αρχική τιμή στο min τη τιμή του 1ου αριθμού που διαβάζουμε.

Γιατί i = 2;

Ένθετοι βρόγχοι

- Ένας βρόγχος μπορεί αν συμπεριλαμβάνει και άλλες εντολές, καθώς επίσης και άλλους βρόγχους (βρόχοι μέσα σε βρόγχο).
- Σε αυτές τις περιπτώσεις θα πρέπει να είμαστε προσεκτικοί στους δείκτες που χρησιμοποιούνται στους βρόγχους.

```
main()
{
int i,j,n;

for( i=1; i<=10; i++ ){
        for( j=1; j<=10; j++ )
                printf("%d\t",i*j);
        printf("\n");
}
```

Προπαίδεια 1-10

Τι θα άλλαζε για τη προπαίδεια 1-5;

```
for( i=1; i<=4; i++ ){
        for( j=1; j<=5; j++ )
                printf("*");
}
```

Θα τυπωθούν 20 (4x5) αστερίσκοι.

Είναι απαραίτητος ο ένθετος βρόγχος;

Ένθετοι βρόγχοι -παραδείγματα

```
int i,j,n;
printf("Dwste n: "); scanf("%d", &n);
for( i=1; i<=n; i++ ){
        for( j=1; j<=i; j++ )
                printf("*");
        printf("\n");
}
```

Έξοδος

```
Dwste n: 5
*
**
***
****
*****
```

Αν αντί για * βάλουμε i?

Αν αντί για * βάλουμε j?

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

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

Ένθετοι βρόγχοι -και άλλο παράδειγμα

```
printf("Please give n: ");
scanf("%d", &n);
for (i = 1; i <= n; i++)
{
    for( j=1; j<=i; j++ )
            printf("*");
    for (j=i;j<=2*n-i;j++)
            printf (" ");
    for (j=1;j<=i;j++)
            printf ("*");
    printf("\n");
}
```

Έξοδος

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

Ένθετοι βρόγχοι -και άλλο παράδειγμα

```
printf("Please give n: ");
scanf("%d", &n);
for (i = 1; i <= n; i++) {
    for (j=1;j<=n-i;j++)
                printf (" ");
    for (j=1;j<=i;j++)
                printf ("*");
    if (i != 1)
        for( j=2; j<=i; j++ )
                    printf("*");
    printf("\n");
}
```

Έξοδος (n=5)

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

Λάθη με τους ένθετους βρόγχους

```
printf("Please give n: ");
scanf("%d", &n);
for( i=1; i<=n; i++ ){
        for( j=1; i<=n/2; j++ )
                printf("j");
        printf("\n");
}
```

```
printf("Please give n: ");
scanf("%d", &n);
for( i=1; i<=n; i++ ){
        for( j=1; j<=n/2; i++ )
                printf("j");
        printf("\n");
}
```