



Προγραμματισμός I

ΕΠΑΝΑΛΗΨΗ - ΒΡΟΧΟΙ

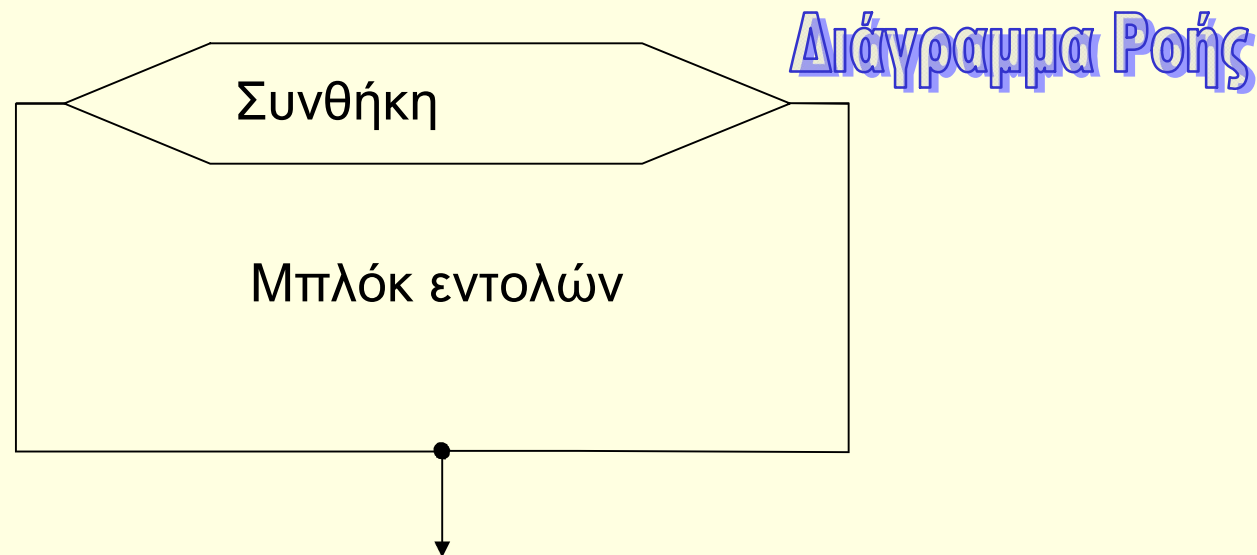
Τμήμα Πληροφορικής & Επικοινωνιών
Δρ. Θεόδωρος Γ. Λάντζος
<http://www.teiser.gr/icd/staff/lantzog>
lantzog@teiser.gr

Βρόχοι – Επανάληψη (Loop)

- Η επανάληψη ενός συνόλου εντολών για όσες φορές επιθυμούμε ή μέχρι να πληρωθεί μια συνθήκη τερματισμού.
- Η επανάληψη αποτελεί ένα δυναμικό εργαλείο για την σχεδίαση και επίλυση σύνθετων προβλημάτων που εμπεριέχουν επαναλαμβανόμενες διεργασίες
- Όταν ο βρόχος θα εκτελείται αενάως καλείται ατέρμων βρόχος (infinite loop)
- Επανάληψη και καθημερινά προβλήματα, π.χ. Ανσασέρ, Αυτόματες μηχανές πώλησης, ΑΤΜ,

Κατηγοριοποίηση Επαναλήψεων

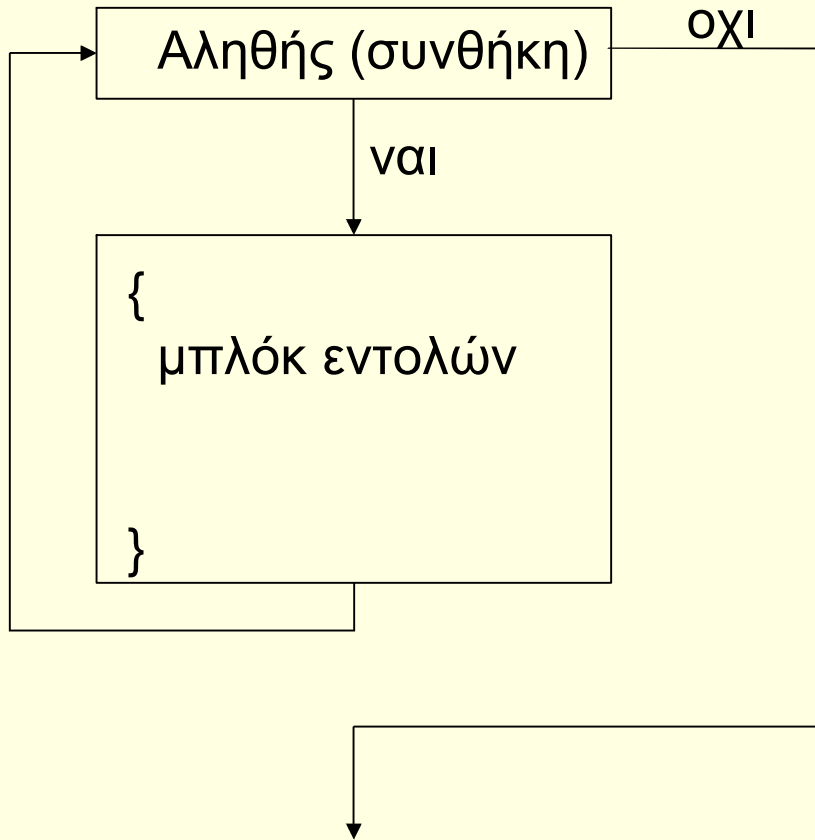
- Α. Με τον τρόπο οδηγήσεως
 - Βρόχους οδηγούμενους από μετρητή
 - Βρόχους οδηγούμενους από γεγονός
- Β. Με την θέση στην οποία γίνεται ο έλεγχος κριτηρίου
 - Με συνθήκη εισόδου (pre-test loops)
(Το κριτήριο τερματισμού ελέγχεται στην αρχή του βρόχου)
 - Με συνθήκη εξόδου (post-test loops)
(Το κριτήριο τερματισμού ελέγχεται στην αρχή του βρόχου)



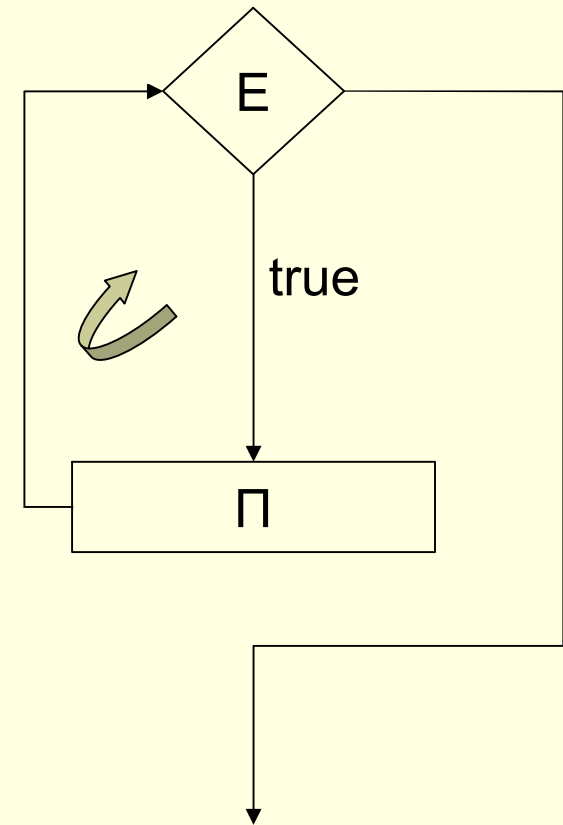
Βρόχος while-do

- Βρόχος με συνθήκη εισόδου
- Μπορεί να οδηγείται τόσο από μετρητή όσο και από γεγονός
- Για όσο η συνθήκη είναι αληθής επανέλαβε
- Έλεγχος συνθήκης στην αρχή
- Ο βρόχος while-do για να έχει λογική υπόσταση θα πρέπει
 - Α. πριν την πρώτη επανάληψη η συνθήκη να είναι αληθής
 - Β. στο μπλόκ εντολών επανάληψης να υπάρχει η δυνατότητα αλλαγής της συνθήκης σε ψευδής
- Επιλέγεται σε περιπτώσεις που δεν είναι γνωστός εκ των προτέρων ο αριθμός των επαναλήψεων

Βρόχος while-do



Μπλόκ Διάγραμμα

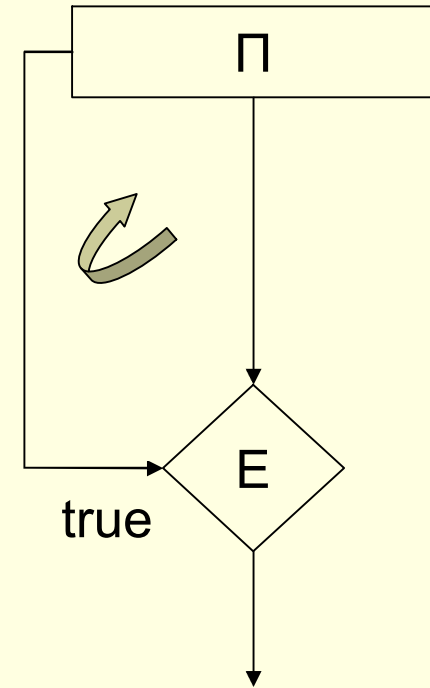
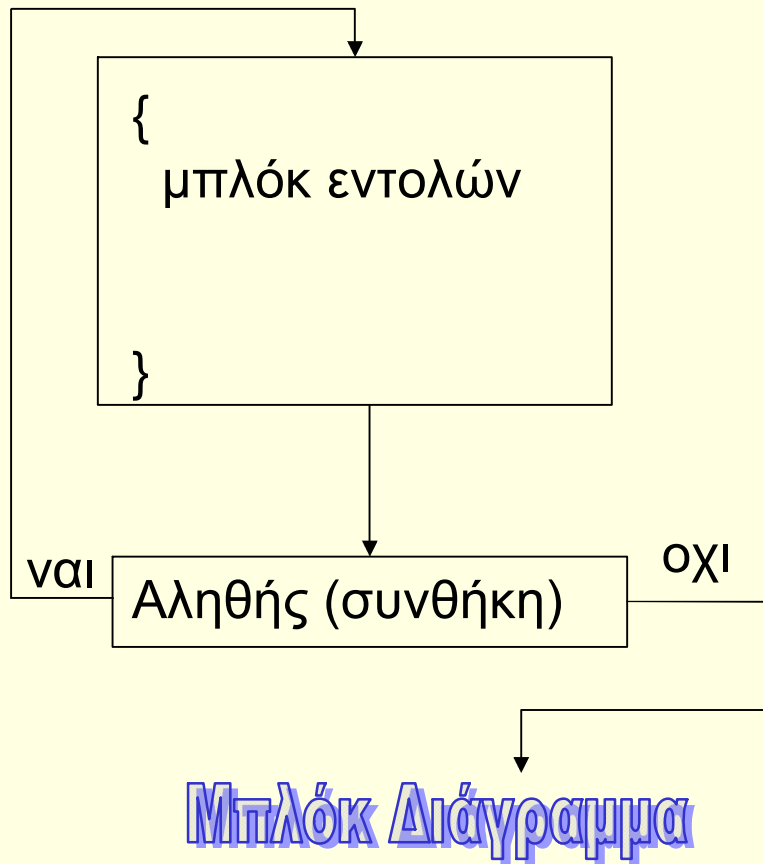


Διάγραμμα ροής

Βρόχος do-while

- Βρόχος με συνθήκη εξόδου
- Δυνατότητα καθοδήγησης με μετρητή ή γεγονός
- Διαφοροποιείται από την while-do στο σημείο ελέγχου της συνθήκης τερματισμού καθώς και στο ότι επιτρέπει την εκτέλεση της πρώτης επανάληψης πριν προχωρήσει στον έλεγχο της συνθήκης
- Δεν προαπαιτεί να είναι η συνθήκη αληθής πριν από την εκτέλεση του βρόχου.
- Πρέπει βέβαια να υπάρχει δυνατότητα μέσα στο μπλόκ εντολών να μετατρέπεται η συνθήκη σε ψευδής για να μπορεί να τερματισθεί ο βρόχος

Βρόχος do-while



Διάγραμμα ροής

Βρόχος do-while στην C – Δήλωση & λειτουργία

```
do  
{  
    προτάσεις, μέσα στις οποίες θα αλλάζει η συνθήκη;  
}  
while (συνθήκη);
```

```
int count=30;  
int limit = 40;  
do  
{  
    count++;  
    printf("count is %d\n",count);  
}  
while(count<limit);  
<επόμενη πρόταση>
```

Εκτέλεσε τις προτάσεις
Έλεγε τη συνθήκη
Εάν είναι αληθής
Ξεκίνησε από την αρχή
Αλλιώς σταμάτησε

παρατηρήσεις

Εκτελείται τουλάχιστον μια φορά

Χαμηλό λόγο χρήσης

Βρόχος στην γλώσσα προγραμματισμού C

■ Εντολή while με σύνταξη

```
while(συνθήκη)
```

```
{
```

```
    προτάσεις, μέσα στις οποίες θα αλλάζει η συνθήκη
```

```
}
```

Έλεγε τη συνθήκη

Εάν είναι αληθής

Προχώρησε στις προτάσεις

Ξεκίνησε από την αρχή

Αλλιώς σταμάτησε

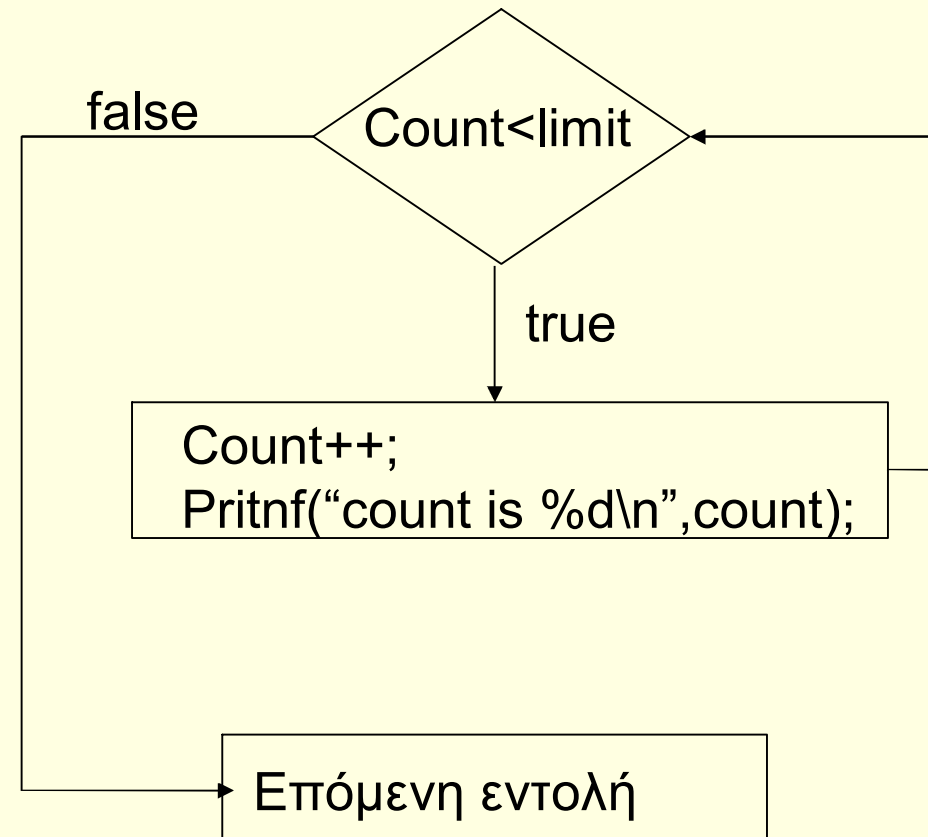
Εάν το σώμα του βρόχου αποτελείται από μια πρόταση, δεν απαιτούνται { }

Παράδειγμα 5.1

```
#include <stdio.h>
```

```
Main()
```

```
{  
  int count=30;  
  int limit = 40;  
  while(count < limit)  
  {  
    count++;  
    printf("count is %d\n", count);  
  }  
  ...  
}
```



Συνθήκη με προπερευόμενο ή παρελκόμενο τελεστη

```
While(++count < 12)
{
    Π1;
    Π2;
}
```

προθεματική σημειογραφία

```
While(count++ < 12)
{
    Π1;
    Π2;
}
```

μεταθεματική

Πού μετατρέπεται η συνθήκη σε ψευδές;

Ποια η διαφορά προθεματικής και μεταθεματικής σημειογραφίας στα loop;

Βρόχος οδηγούμενος από μετρητή

- Χρησιμοποιεί έναν μετρητή
- Έχει αρχική συνθήκη
- Οδηγείται από μετρητή (ο μετρητής μέλος στην συνθήκη)
- Έχει Δυνατότητα ρύθμισης προσαύξησης θετική – αρνητική
- Παρέχει την Δυνατότητα διαβάθμισης προσαύξησης (βήμα)

Βρόχος οδηγούμενος από μετρητή στην C

```
For( αρχική τιμή μετρητή; Συνθήκη; βήμα μετρητή)  
{  
    προτάσεις;  
}
```

Αρχικοποίησε το μετρητή
Έλεγε τη συνθήκη
Εάν είναι αληθής
 εκτέλεσε τις προτάσεις
 ενημέρωσε το μετρητή
 Επανήλθε στον έλεγχο της συνθήκης
Αλλιώς ενημέρωσε το μετρητή και σταμάτησε

Όταν οι προτάσεις ή εκφράσεις ή εντολές είναι μόνο μία
τα άγκιστρα μπορούν να παραλειφθούν

Εντολή For παραδείγματα

```
For(n=0;n<10;n++) printf("\n=%d\n",n);
```

```
For(n=10;n>0;n--) printf("n=%d\n",n);
```

```
For(n=0;n<60;n=n+13) printf("n=%d\n",n);
```

```
For(n='a';n<'z';n++) printf("n=%d, the ASCII value is %d\n",n,n);
```

```
For(n=0;n<60.0; n=1.2*n) printf("n=%f\n",n);
```

Εντολή For παραδείγματα

5.3

```
#include <stdio.h>
Main()
{
  int count=30;
  int mac_count = 40;
  for(count=0; count < max_count; count++)
  {
    printf("count is %d\n",count);
    <άλλες προτάσεις>
  }
  ...
}
```

5.4

```
#include <stdio.h>

main()
{
  int x,y;
  for(x=0,y=0; x+y<100; x=x+20, y=y+10)
    printf("x=%d\ty=%d\tx+y=%d\n",x,y,x+y);
  printf("x=%d\ty=%d",x,y);
}
```

Τελεστής ‘,’ και χρήση σε loop

- Επιτρέπει την παράθεση περισσότερων της μίας έκφρασης σε θέσεις όπου επιτρέπεται μία έκφραση. Η τιμή της έκφρασης είναι η τιμή της δεξιότερης των εκφράσεων
- Έχει περιορισμένη εφαρμογή
- Στην for χρησιμοποιείται ως συνθετικό των εκφράσεων αρχικοποίησης και ανανέωσης

```
#include <stdio.h>

main()
{
    int i, j;

    for(i=0, j=10; i<8 ; i++, j++)
    {
        printf("\ni=%d\t j=%d",i,j);
    }
    getchar();
}
```

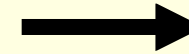

Μετασχηματισμός βρόχων for - while

for loop

```
for(αρχική τιμή μετρητή; Συνθήκη ; Βήμα μετρητή)
{
    προτάσεις;
}
```

while loop

```
αρχική τιμή μετρητή;
while(Συνθήκη )
{
    προτάσεις;
    βήμα μετρητή;
}
```



Παράδειγμα 5.5

```
for (n=4; n>0; n--)
{
    printf(“%d,”,n/2);
}
```



```
n=4;
while (n>0)
{
    printf(“%d,”,n/2);
    n--;
}
```

Μετασχηματισμός βρόχων for σε do while

```
# include <stdio.h>
```

```
main()
```

```
{  
  int n;  
  for(n=4; n>1; n--)  
    printf(“%d,”,10%n);  
}
```



```
# include <stdio.h>
```

```
main()
```

```
{  
  int n;  
  n=4;  
  do  
  {  
    printf(“%d,”,10%n);  
    n--;  
  }  
  while(n-1);  
}
```

Ένθετοι βρόχοι

- Αναφέρονται ως και 'φωλιασμένος βρόχος'
- Ονομάζεται ο βρόχος που περικλείεται σε έναν άλλο
- Ο εσωτερικός βρόχος λογίζεται ως μια πρόταση του εσωτερικού – δηλαδή πρώτα θα εκτελείται ο εσωτερικός βρόχος και μετά η επόμενη επανάληψη του εξωτερικού
- Η C επιτρέπει πολλαπλή ένθεση
- Ο συνολικός αριθμός επαναλήψεων σε έναν πολλαπλό βρόχο είναι το γινόμενο του αριθμού των επαναλήψεων όλων των επιμέρους βρόχων

Παραδείγματα ένθετων βρόχων

5.8

```
for(i=0;i<4;i++)
{
  for(j=0;j<3;j++)
  {
    printf("(%d.%d)",i,j);
  }
}
printf("\n");
```

5.9

```
#include <stdio.h>
void main()
{
  int l,j,k,temp;
  printf(" l i^2 l^3 i^4\n");
  for(i=1;i<10;i++)
  {
    for(j=1;j<5;j++)
    {
      temp=1;
      for(k=0;k<j;k++)
        temp=temp*i;
      printf("%9d",temp);
    }
    printf("\n");
  }
}
```

Διακοπτόμενη βρόχοι στη C

- Ονομάζεται ο βρόχος στον οποίο η επανάληψη διακόπτεται για μεταφορά του ελέγχου α. στην αρχή του βρόχου και έλεγχο της συνθήκης β. εκτός βρόχου

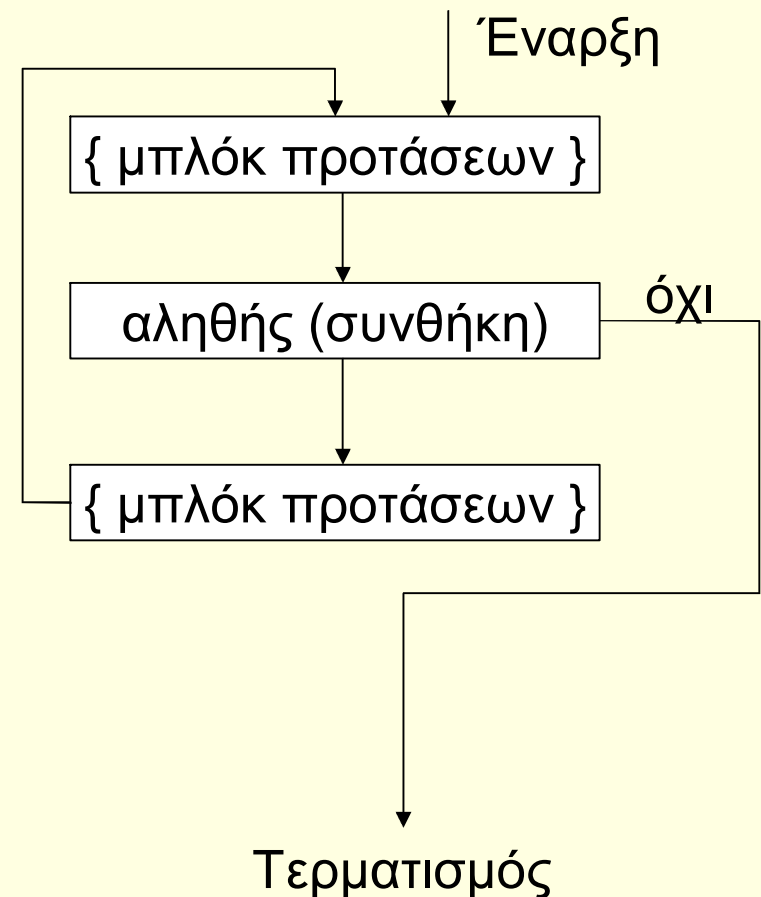
Υλοποίηση με την χρήση

- Κωδικής λέξης break
- Πρόταση continue
- Πρόταση goto

Break

- Στους βρόχους η κωδική λέξη `break` έχει την σημασία του τερματισμού εκτέλεσης ενός βρόχου, μεταβιβάζοντας τον έλεγχο του προγράμματος στην εντολή που βρίσκεται αμέσως μετά το βρόχο

```
#include <stdio.h>
void main()
{
    int t;
    for(t=0;t<100;t++)
    {
        printf("\n%d",t);
        if(t==10) break;
    }
    printf("\nt=%d",t);
}
```



Πρόταση continue

- Μεταφέρει τον έλεγχο της ροής στην αρχή του βρόχου, παραλείποντας την εκτέλεση του υπολοίπου τμήματος του σώματος του βρόχου και προχωρώντας στην επόμενη επανάληψη
- Στις εντολές while και do-while η εντολή continue υποχρεώνει τον έλεγχο του προγράμματος να περάσει κατευθείαν στην συνθήκη ελέγχου και να προχωρήσει κατόπιν στην επεξεργασία του βρόχου
- Αντιθέτως στην εντολή for ο υπολογιστής εκτελεί πρώτα το τμήμα του βρόχου (αύξηση μετρητή) και κατόπιν τη συνθήκη ελέγχου

5.11

```
#include <stdio.h>

void main()
{
    int x;
    for(x=0;x<100;x++)
    {
        if(x%2) continue;
        printf("\t%d",x);
    }
    getch();
}
```

Η πρόταση goto

- Είναι η πρόταση ρητής διακλάδωσης
 - goto <ετικέτα>
- Μεταφέρει τον έλεγχο στην πρόταση που σημειώνεται με την ετικέτα ως
 - <ετικέτα>:πρόταση
- Αποφυγή χρήσης διότι διασπά τις αρχές του δομημένου προγραμματισμού δημιουργώντας επιπλοκές στην λογική ροή του προγράμματος
- Συνήθως χρησιμοποιείται όταν θέλουμε να διασπάσουμε το τμήμα του προγράμματος που εκτελείται μέσα σε πολλές ένθετες επαναλήψεις (βρόχους)

Κανόνες χρήσης προτάσεων επανάληψης

- Σωστή χρήση στηλοθεσίας για εύκολη αναγνωσιμότητα κώδικα
- Χρησιμοποιείται {} για να περικλείετε περισσότερες της μίας πρότασης
- Αποφεύγετε τη χρήση της πρότασης διακλάδωσης goto
- Προτιμήστε το βρόχο επανάληψης με συνθήκη εισόδου (while) από τον αντίστοιχο με συνθήκη εξόδου (do-while)
- Αποφεύγετε κατά το δυνατόν τη χρήση των break και continue σε βρόχους επανάληψης
- Εκτελέστε λεπτομερή έλεγχο και βεβαιωθείτε ότι κάθε συνθήκη βρόχου επανάληψης οδηγεί στην έξοδο μετά από πεπερασμένες επαναλήψεις, έτσι ώστε να μη δημιουργούνται ατέρμονες βρόχοι