

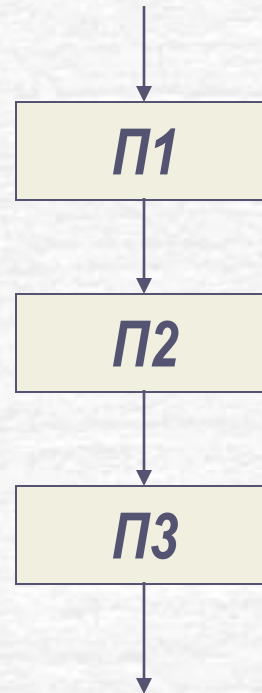


# ΤΕΧΝΟΛΟΓΙΑ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ

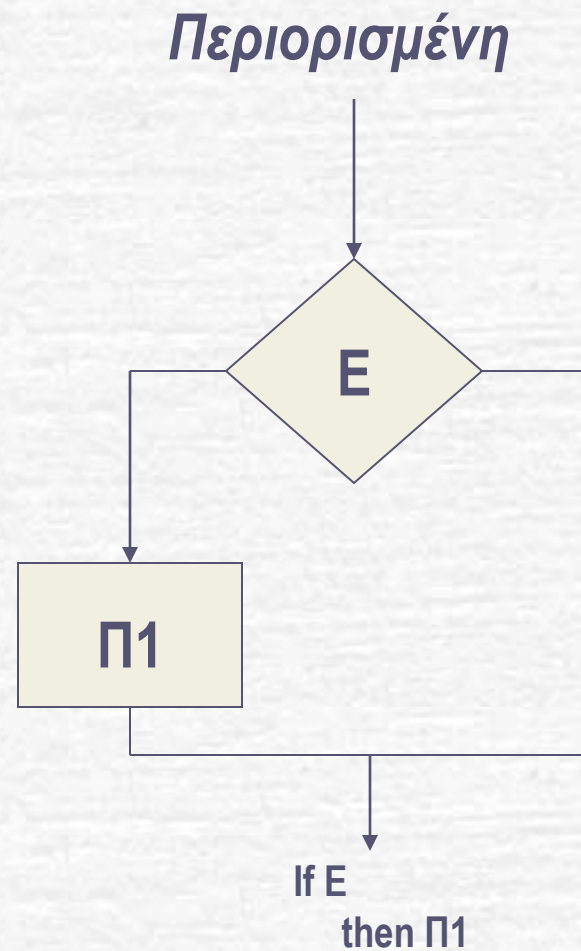
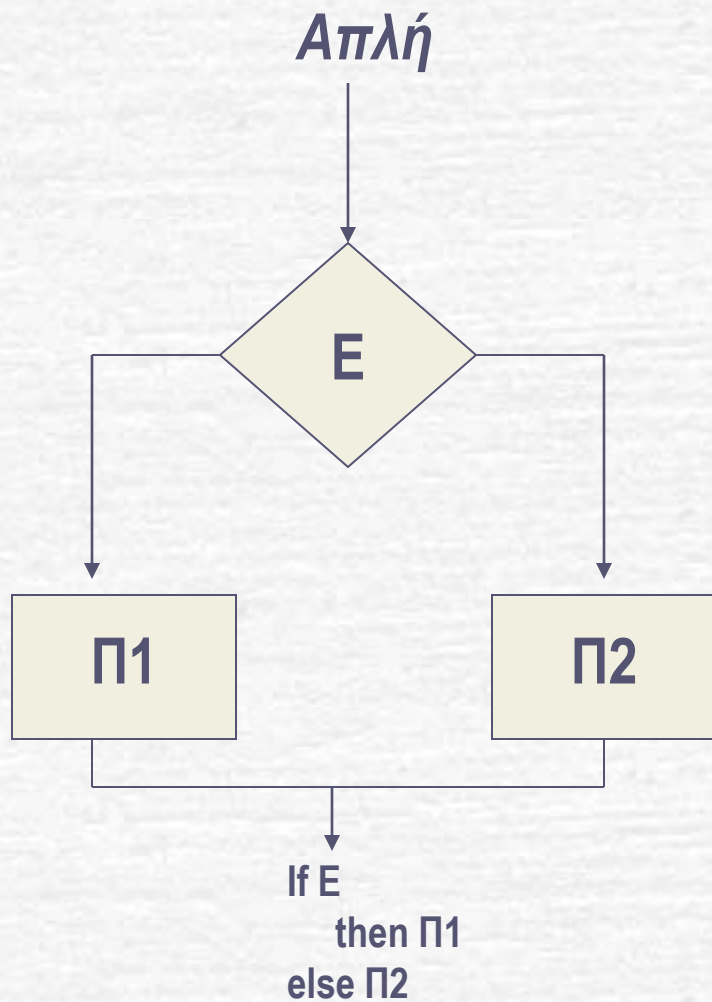
- Έλεγχος ροής
- Δομή επιλογής (if, switch)
- Δομές επανάληψης (while, do-while, for)
- Διακλάδωση χωρίς συνθήκη (break, continue, goto)

# ΕΛΕΓΧΟΣ ΡΟΗΣ

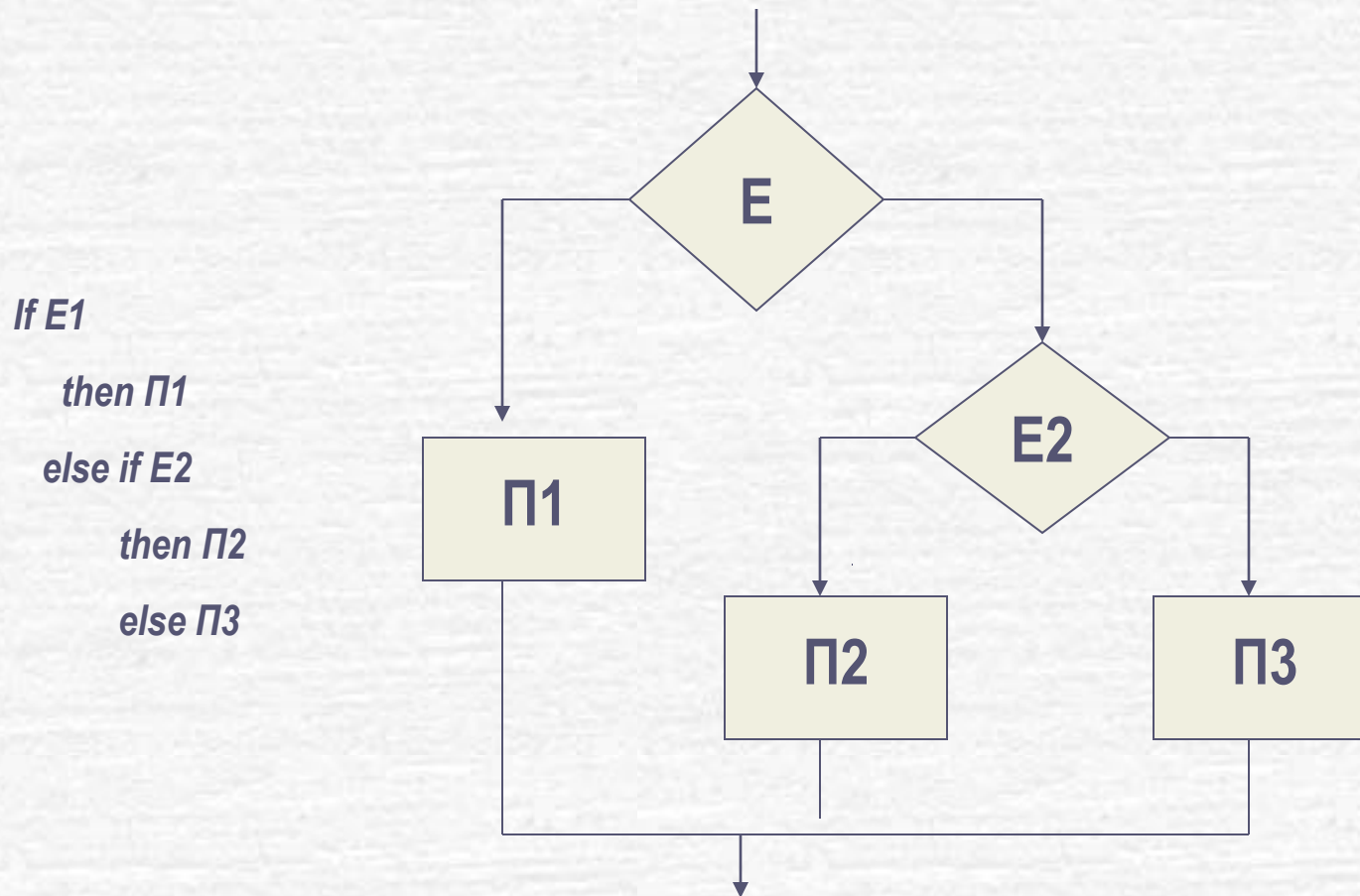
- Αλγοριθμικές δομές
  - ακολουθία
  - επιλογή
  - επανάληψη
- Ακολουθία



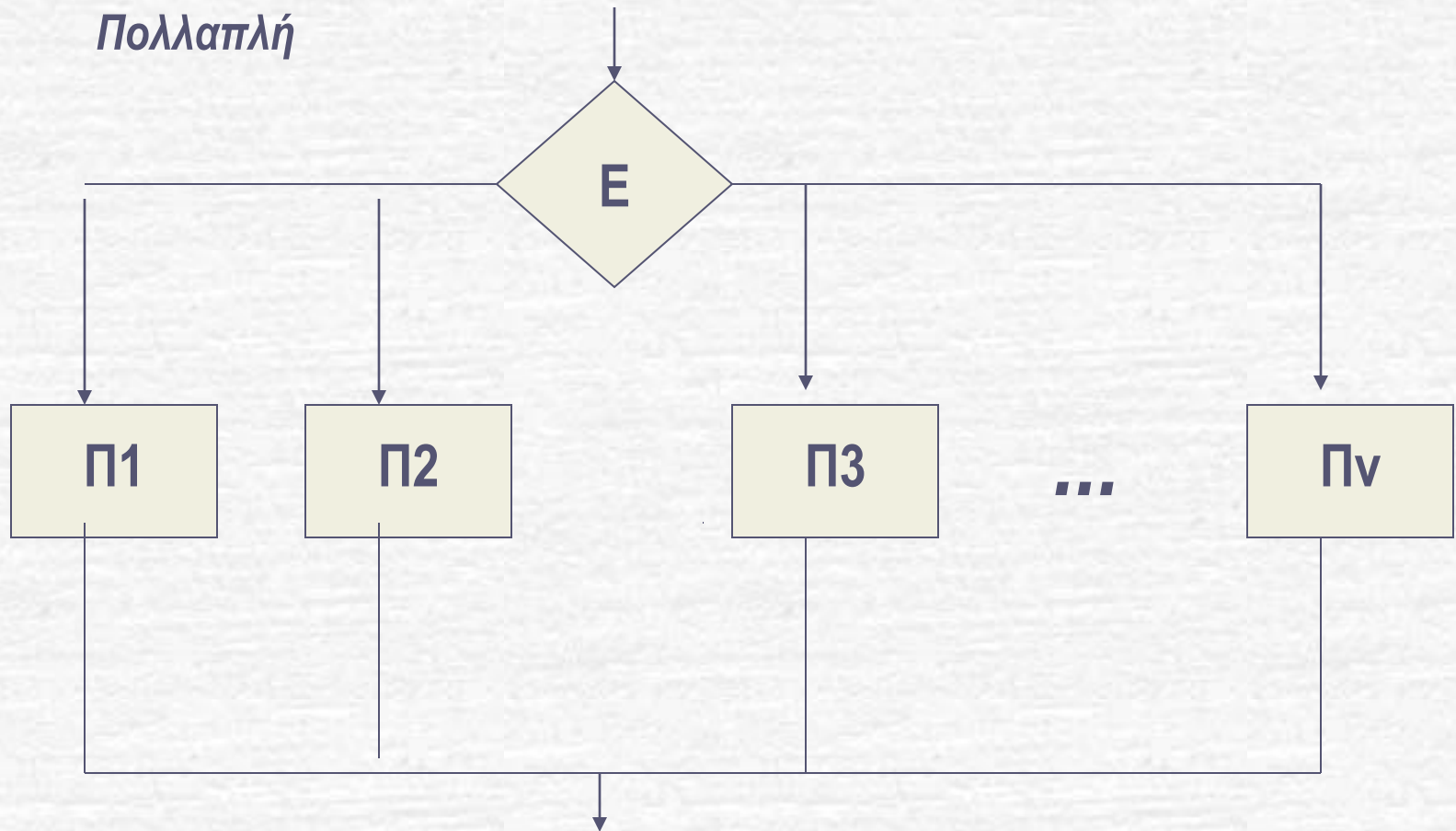
# Δομή Επιλογής (1)



# Δομή Επιλογής (2)



# Δομή Επιλογής (3)



# Επιλογή στη C-if

```
If (<έκφραση>
    <πρόταση1>
[else <πρόταση2>]
```

απλή ή σύνθετη

```
If (<έκφραση>
    <πρόταση1>
    else if (<έκφραση2>)
        <πρόταση2>
    else if (<έκφραση3>)
        <πρόταση3>
    else <πρόταση4>
```

Εμφωλευμένο If

# Παράδειγμα

```
#include <stdio.h>

main ( ) {
float num;
printf("Δώσε αριθμό: ");
scanf("%f", &num);
If (num < 0)
    printf("Η απόλυτη τιμή του %f είναι: %f\n", num, -num);
    else printf("Η απόλυτη τιμή του %f είναι: %f\n", num, num);
printf("Η απόλυτη τιμή του %f είναι: %f\n", num, (num<0)?-num:num);
}
```



# Επιλογή στη C-switch (1)

```
switch (<έκφραση>) {  
    case <σταθ-εκφρ1>: <προτ1>  
    case <σταθ-εκφρ2>: <προτ2>  
        ...  
    case <σταθ-εκφρν> : <προτν>  
    default: <πρόταση>  
}
```

# Επιλογή στη C-switch (2)

## Κανόνες

- Κάθε <σταθ-έκφρι> πρέπει να είναι μία τιμή int ή char ή μία έκφραση μόνο με τέτοιες τιμές
- Δύο <σταθ-εκφρι> δεν μπορεί να έχουν την ίδια τιμή
- Αν <έκφραση>=<σταθ-εκφρχ> τότε εκτελούνται όλες οι παρακάτω της x προτάσεις (χρήση break)
- Η <πρόταση> εκτελείται μόνο όταν καμιά από τις <πρότι> δεν ικανοποιείται
- Η default δεν είναι απαραίτητο να είναι στο τέλος

# Παράδειγμα

```
switch (choice) {  
    case 1:  
        x=a+b;  
        break;  
    case 2:  
        x=a-b;  
        break;  
    case 3:  
        x=a*b;  
        break;  
    case 4:  
        x=a/b;  
        break;  
    default:  
        printf("Ανύπαρκτη επιλογή");  
        break;  
}
```

# Δομές (ή Βρόχοι) Επανάληψης

## Μέρη

- Έκφραση τερματισμού
  - Συνθήκη
  - Μετρητής
- Σώμα επανάληψης

## Τύποι

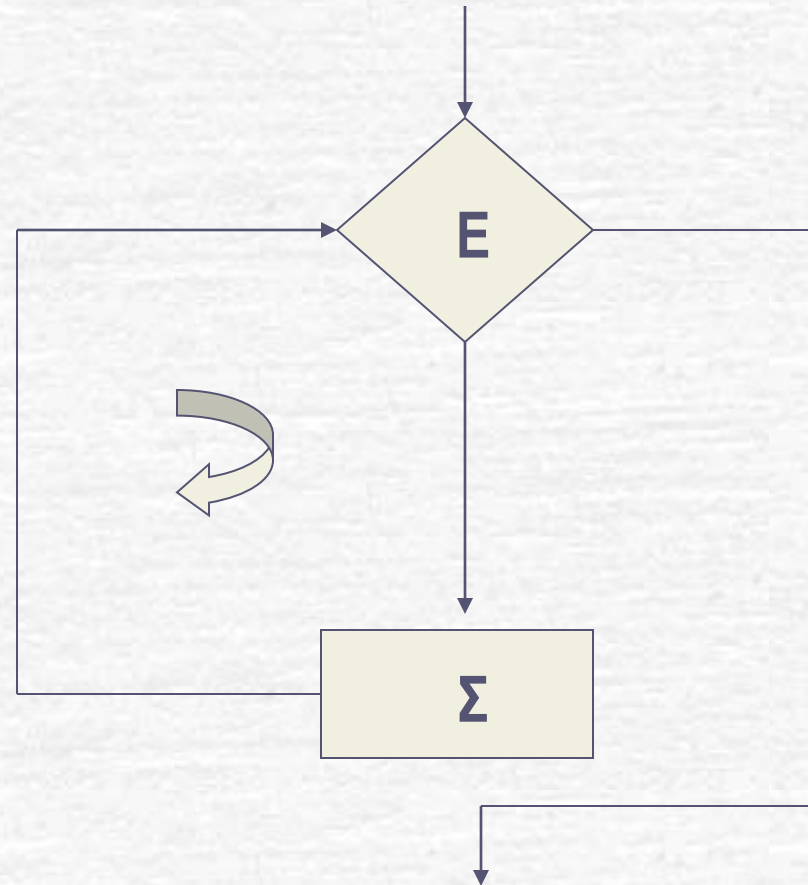
- Δομή επανάληψης με συνθήκη
  - εισόδου
  - εξόδου
- Δομή επανάληψης με μετρητή

# Δομή Επανάληψης με Συνθήκη Εισόδου

While E do Σ (Pascal)

while E (C)

(Το Σ μπορεί να μην  
εκτελεστεί ποτέ)

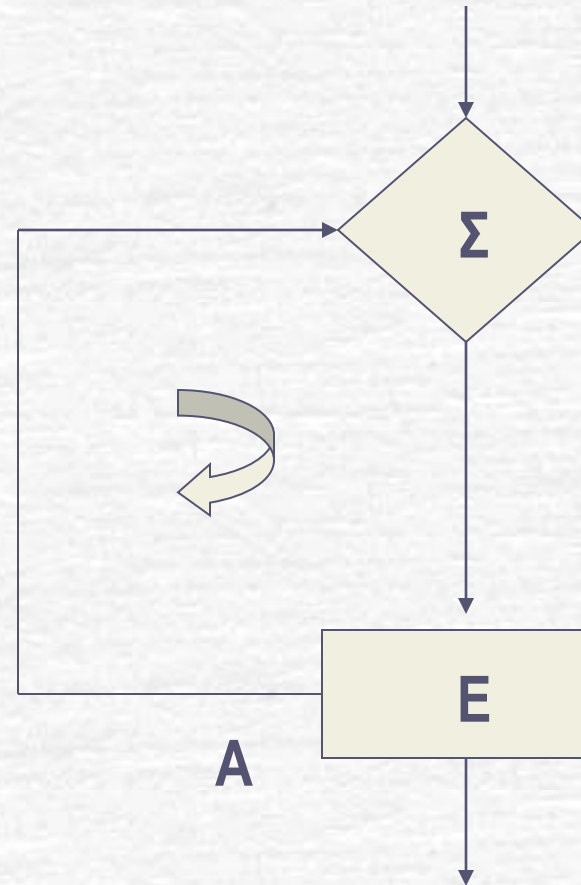


# Δομή Επανάληψης με Συνθήκη Εξόδου

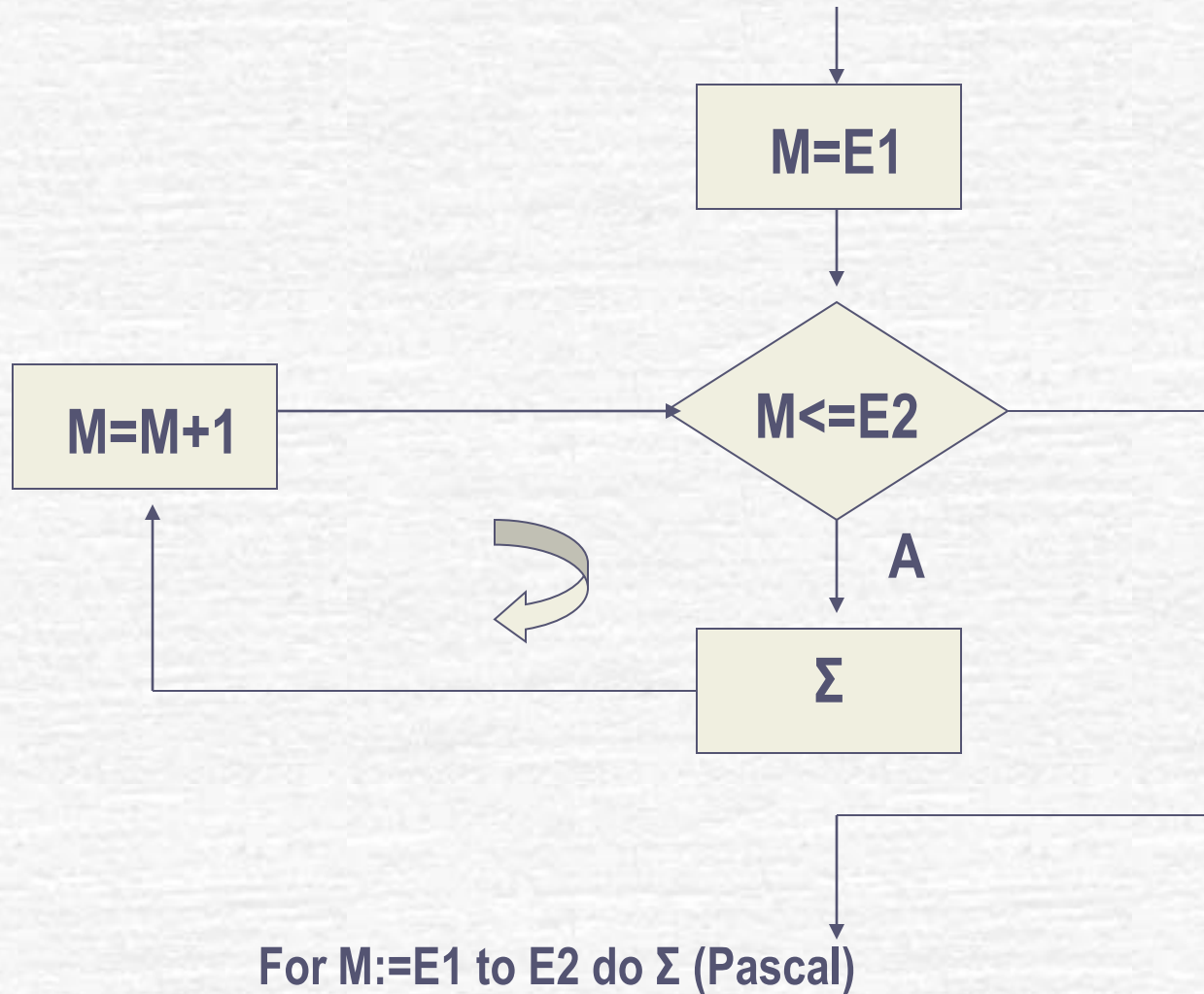
repeat  $\Sigma$  until  $E$  (Pascal)

do  $\Sigma$  while  $E$  (C)

(Το  $\Sigma$  θα εκτελεστεί  
τουλάχιστον μία φορά)



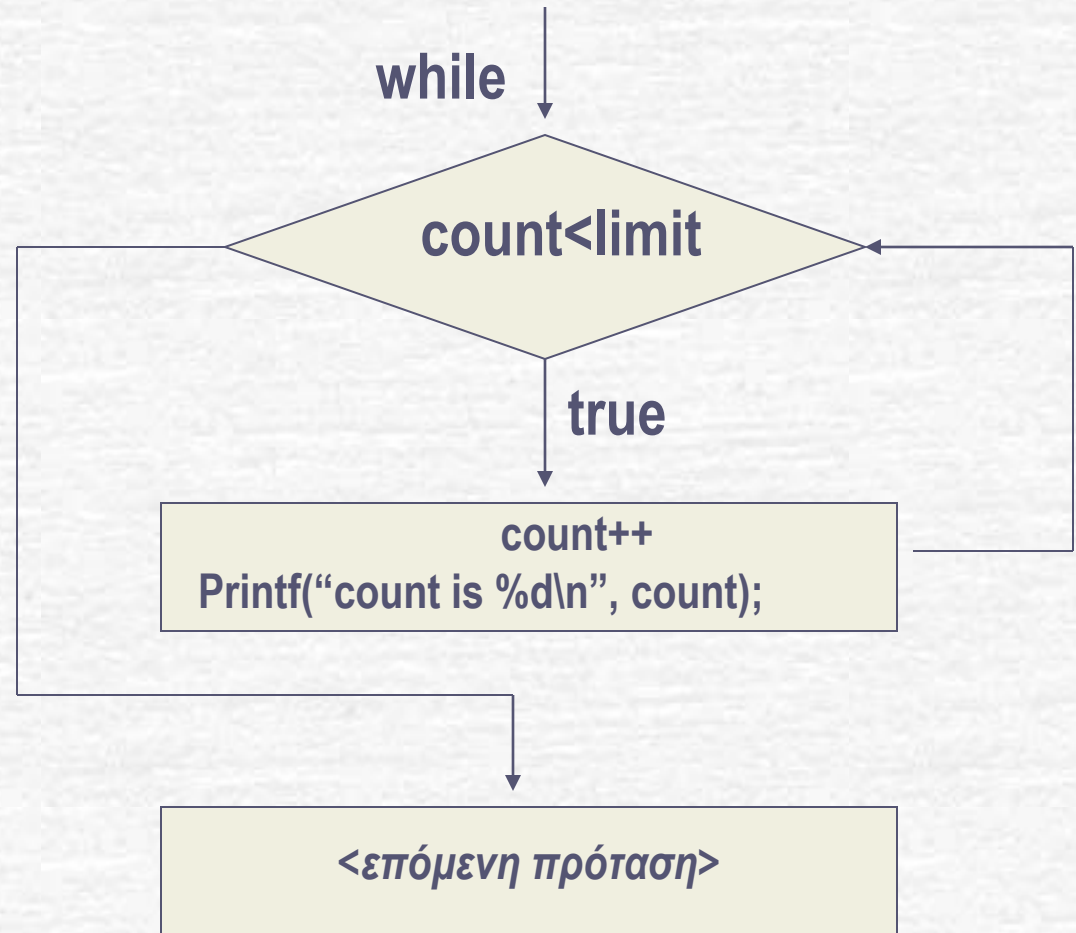
# Δομή Επανάληψης με Μετρητή



# Δομή Επανάληψης στη C: while

**while** (<έκφραση>  
<πρόταση>

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

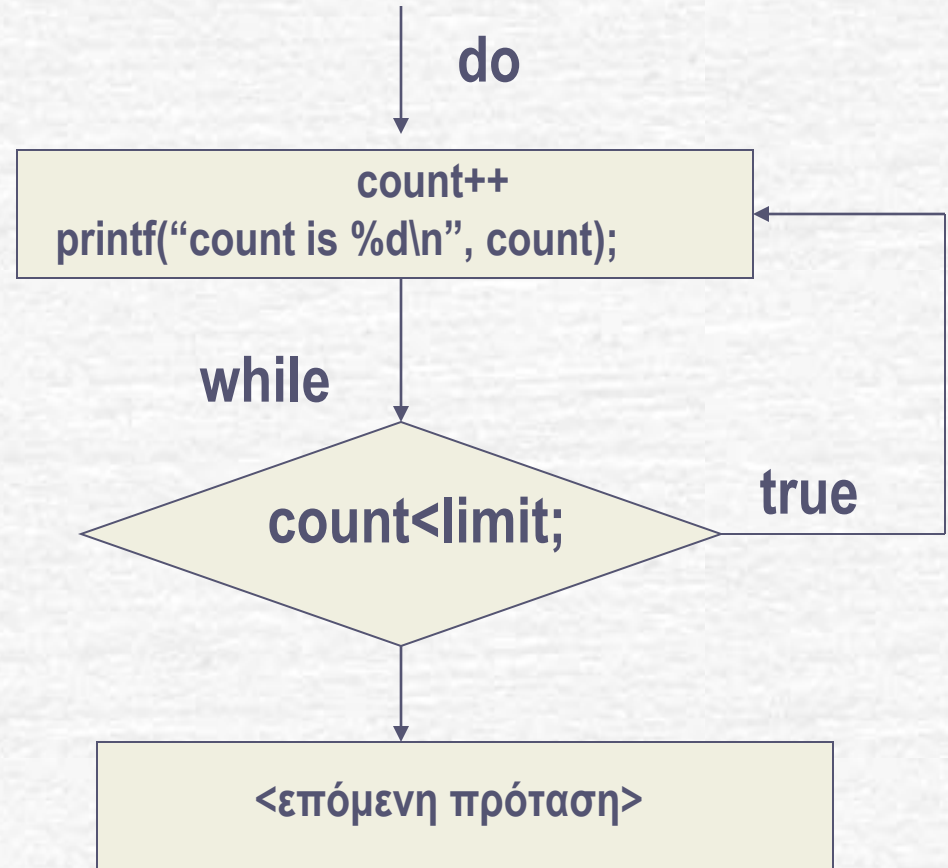




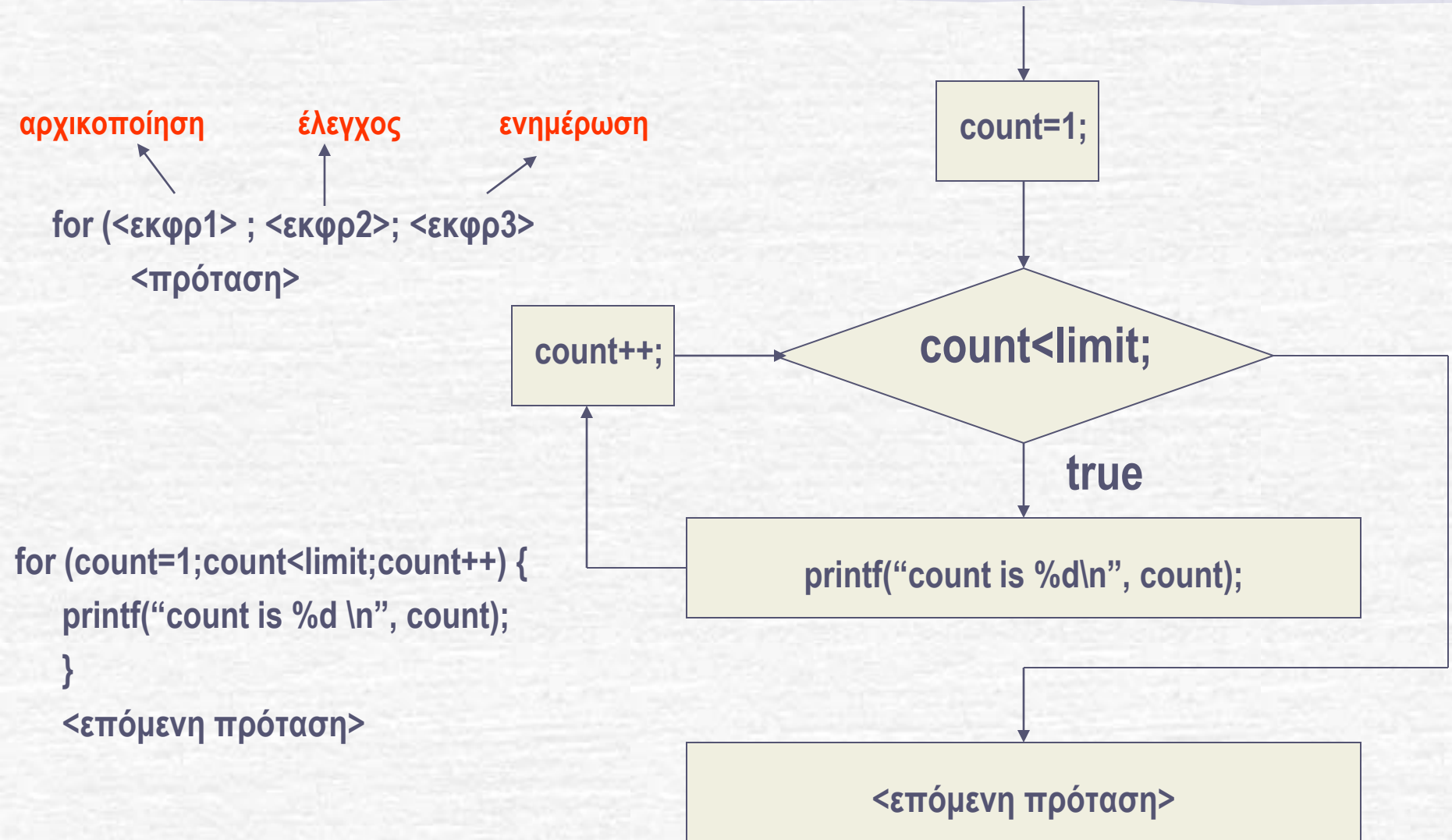
# Δομή Επανάληψης στη C: do while

```
do <επόμενη πρόταση>  
while (<έκφραση>;
```

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



# Δομή Επανάληψης στη C: for

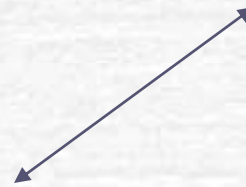


# Ισοδυναμία Δομών Επανάληψης

```
for (e1;e2;e3)  
  Π1
```



```
e1;  
while (e2) {  
  Π1;  
  e3;  
}
```



```
e1;  
do {  
  Π1;  
  e3;  
}  
while (e2)
```

# Επιλογή Δομής Επανάληψης

- Προτιμούμε τη δομή με συνθήκη εισόδου (while) από αυτή με συνθήκη εξόδου (do-while)
- Προτιμούμε την for από την while, αν υπάρχει (ή μπορεί να οριστεί) απαριθμητής που συνοδεύεται από αρχικοποίηση και ανανέωση της τιμής του.

# Διακλάδωση χωρίς συνθήκη

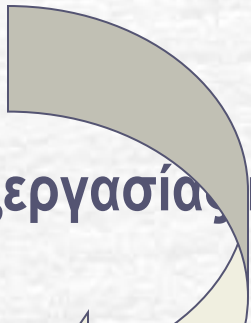
- Διαχείριση ειδικών περιπτώσεων σε προτάσεις επανάληψης
  - break
  - continue
- Ρητή διακλάδωση
  - goto <ετικέτα>

# Πρόταση break (1)

- Προκαλεί την έξοδο μόνο από τον πιο εσωτερικό βρόχο
- **Καταστρέφει τη δόμηση** του κώδικα
- Υπάρχει πάντα τρόπος να γραφεί κώδικας χωρίς τη χρήση της.

# Πρόταση break (2)

```
while (<έκφραση>) {  
    if ειδική περίπτωση {  
        προτάσεις επεξεργασίας ειδικής περίπτωσης  
        break;  
    }  
    προτάσεις επεξεργασίας κανονικών περιπτώσεων  
}
```



# Πρόταση break (3)

```
for (i=0; i<max; i++) {  
    if (num[i]<0)  
        break;  
    :  
}  
printf(.....);
```



```
for (i=0; (i<max) && (num[i]>=0); i++) {  
    :  
    :  
}  
printf(.....);
```

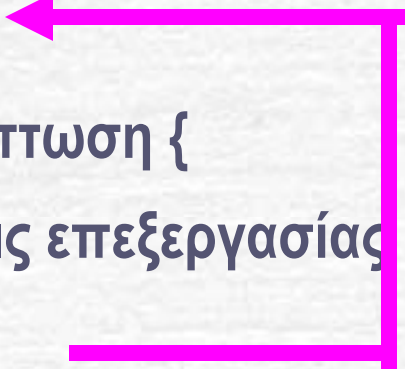


# Πρόταση continue (1)

- Προκαλεί την έναρξη της επόμενης επανάληψης for, while ή do
- Επηρεάζει μόνο το πιο εσωτερικό βρόχο

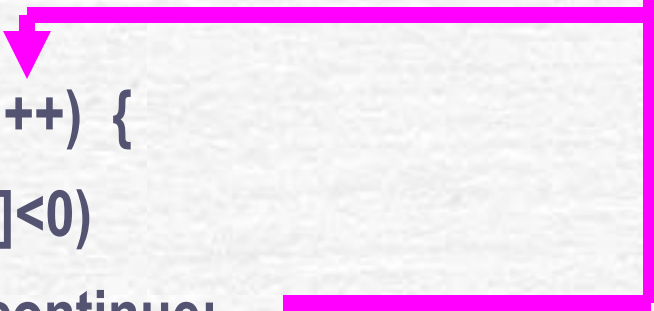
# Πρόταση continue (2)

```
while (<έκφραση>) {  
    if κανονική περίπτωση {  
        προτάσεις επεξεργασίας κανονικής περίπτωσης  
        continue;  
    }  
    προτάσεις επεξεργασίας ειδικών περιπτώσεων  
}
```



# Πρόταση continue (3)

```
for (i=0; i<max; i++) {  
    if (num[i]<0)  
        continue;  
    :  
}
```



```
printf(.....);
```

```
For (i=0; (i<max) && (num[i]>=0); i++) {  
    :  
    :  
}
```

```
printf(.....);
```

# Πρόταση goto

- Καταστρέφει την δόμηση του κώδικα
- Είναι πάντα εύκολο να γράφεις κώδικα χωρίς τη χρήση τους