

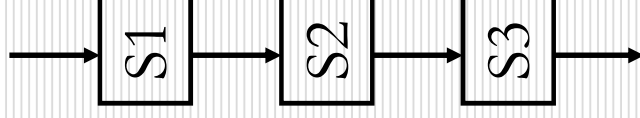
2. Δομές ελέγχου ροής στη C

- Ελέγχουν την ροή εκτέλεσης σε ένα πρόγραμμα

□ Κάθε πρόγραμμα μπορεί να γραφεί χρησιμοποιώντας τρεις δομές ελέγχου ροής :
την ακολουθία, την επιλογή και την επανάληψη

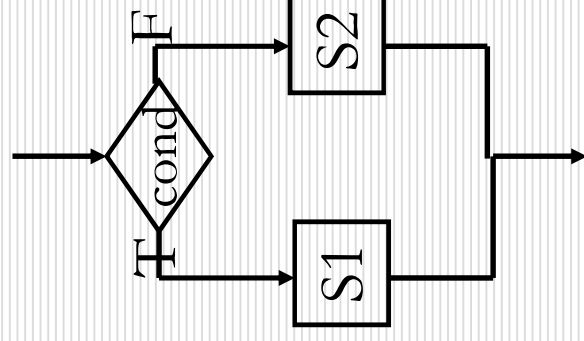
□ Ακολουθιακές δομές

- Οι εντολές εκτελούνται η μια μετά την άλλη με τη σειρά που είναι γραμμένες



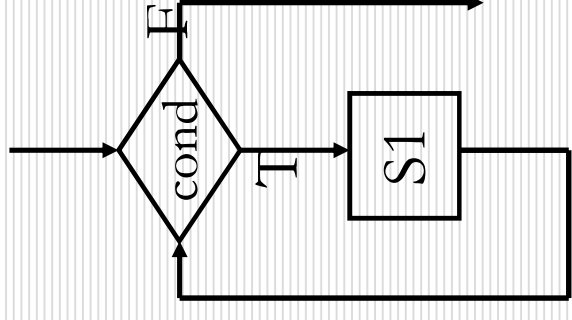
□ Δομές επιλογής

- if
- if...else
- switch



□ Δομές επανάληψης:

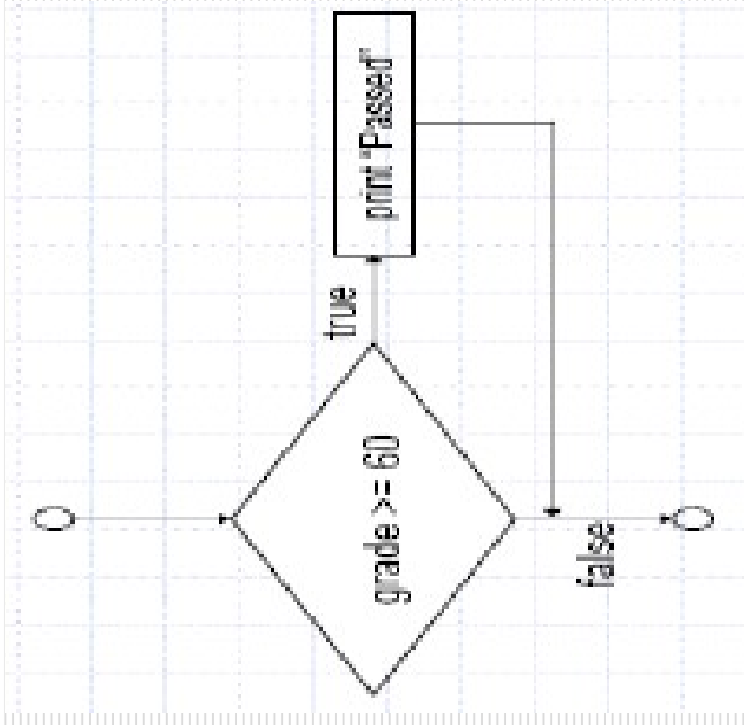
- for
- while
- do...while



□ έχουν μία είσοδο και μία έξοδο

Η εντολή επιλογής if

- Γενική μορφή εντολής if
if (έκφραση)
 εντολή;
- Αν η έκφραση της if είναι αληθής - **true**
 - Εκτελείται η εντολή printf και το πρόγραμμα συνεχίζει με την επόμενη εντολή
- Αν η έκφραση της if είναι ψευδής - **false**
 - Η εντολή printf αγνοείται και το πρόγραμμα συνεχίζει με την επόμενη εντολή



```
if ( grade >= 60 )  
    printf( "Passed\\n" );
```

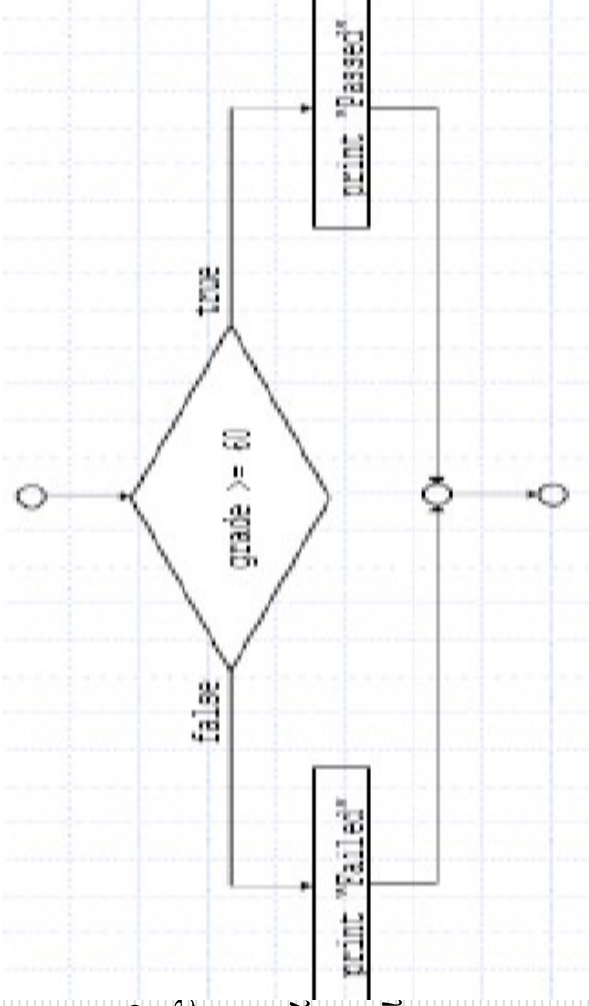
Παράδειγμα: if

```
#include <stdio.h>

void main() {
    int num;
    printf ("Enter an integer: \n");
    scanf ("%d", &num);
    if (num < 0)
        printf ("Number is negative.\n");
    if (num >= 0)
        printf ("Number is non-negative.\n");
}
```

Η εντολή επιλογής if...else

- **if**
 - Μια ενέργεια πραγματοποιείται μόνο αν η έκφραση της if είναι true
- **if...else**
 - Προσδιορίζει μια ενέργεια που πρέπει να εκτελεστεί τόσο όταν η έκφραση είναι true όσο και όταν η έκφραση είναι false



- **Γενική μορφή εντολής if...else**

```
if (έκφραση)
    εντολή1;
else
    εντολή2;
```

```
if ( grade >= 60 )
    printf( "Passed\n");
else
    printf ( "Failed\n");
```

Τριαδικός τελεστής (:?)

- παρέχει απλά έναν εναλλακτικό τρόπο γραφής της εντολής if.
- **συνθήκη ? εντολή if true : εντολή if false**
- `grade >= 60 ? printf("Passed\n") : printf("Failed\n");`

Παράδειγμα: if...else

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

main() {
    double num;

    printf ("Δώσε ένα θετικό αριθμό:");
    scanf ("%lf", &num);
    if (num<0)
        printf ("Λάθος εισόδος: Αριθμός αρνητικός\n");
    else
        printf ("Η τετραγωνική ρίζα του %lf είναι %lf\n", num, sqrt(num));
}
```

‘Ενθετες/ «Φωλιασμένες» (nested) εντολές if...else

```
...  
if (grade >= 90)  
    printf ("A\n");  
else if (grade >= 80)  
    printf ("B\n");  
else if (grade >= 70)  
    printf ("C\n");  
else if (grade >= 60)  
    printf ("D\n");  
else  
    printf ("F\n");  
...
```

- ❑ Ποια if σχετίζεται με ποια else ;
- ❑ Μια else σχετίζεται πάντα με την πλησιέστερη if που υπάρχει στο ίδιο τμήμα κώδικα και δεν είναι συσχετισμένη με μια άλλη if

Κανόνες της switch

- ❑ Κάθε case πρέπει να έχει μία int ή char σταθερά ή μια σταθερά έκφραση
- ❑ Δύο case δεν μπορούν να έχουν την ίδια τιμή
- ❑ Η πρόταση της default εκτελείται όταν καμία από τις case δεν ικανοποιείται

Παράδειγμα: switch

```
#include <stdio.h>

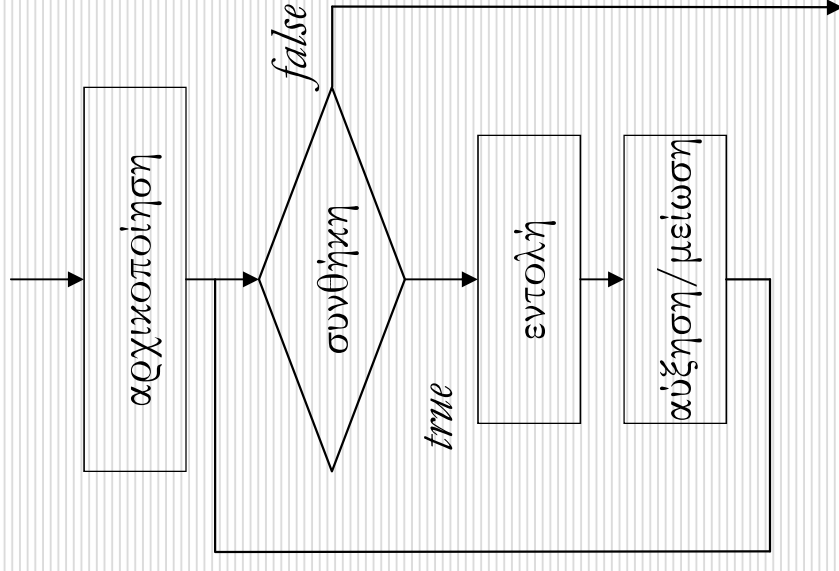
void main() {
    int day;
    printf ("Dose arithmo hmeras (1-7):"); scanf ("%d", &day);
    switch (day) {
        case 1: printf ("Monday \n"); break;
        case 2: printf ("Tuesday \n"); break;
        case 3: printf ("Wednesday \n"); break;
        case 4: printf ("Thursday \n"); break;
        case 5: printf ("Friday \n"); break;
        case 6: printf ("Saturday \n"); break;
        case 7: printf ("Sunday \n"); break;
        default: printf ("Edoses lathos arithmo\n");
    }
}
```

Η εντολή επανάληψης for

- ❑ Χρησιμοποιείται όταν ο αριθμός των επαναλήψεων είναι γνωστός
- ❑ Γενική μορφή εντολής for

for (αρχικοποίηση; συνθήκη-ελέγχου; αύξηση/μείωση εντολή;

- ❑ **Αρχικοποίηση**
 - δίνει αρχική τιμή στη μεταβλητή που ελέγχει το βρόχο εκτελείται μόνο μια φορά πριν ξεκινήσει ο βρόχος
- ❑ **Συνθήκη-ελέγχου**
 - Αν το αποτέλεσμα είναι true ο βρόχος επαναλαμβάνεται. Αν το αποτέλεσμα είναι false ο βρόχος σταματά.
- ❑ **Αύξηση/ μείωση**
 - εκτελείται στο τέλος του βρόχου



Παράδειγμα: for

```
int sum=0,i;  
for (i=1; i<=100; i++)  
    sum = sum + i*i;
```

$$\sum_{i=1}^{100} i^2$$

□ Αρχικοποίηση και αύξηση/ μείωση

- Μπορούν να αποτελούνται από μια ομάδα εντολών που διαχωρίζονται με κόμμα (,)

```
for (int i = 0, j = 0; i + j <= 10; i++, j++)  
    printf( "%d \n", i + j );
```

Παράδειγμα: for

□ Υπολογισμός του $3^2+6^2+\dots+18^2$

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

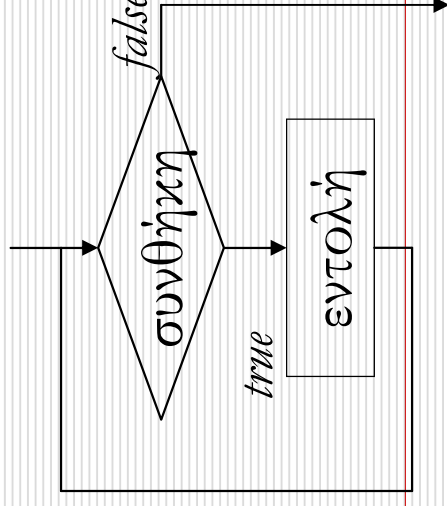
```
void main() {
    double sum=0; int i;
    for ( i=0; i<=20; i=i+3)
        sum += pow(i,2);
    printf("SUM= %lf \n", sum);
}
```

Η εντολή επανάληψης while

- Χρησιμοποιείται όταν ο αριθμός των επαναλήψεων δεν είναι γνωστός
- Χρειάζεται
 - Μια μεταβλητή ελέγχου (ή μετρητής βρόχου)
 - Αρχικοποίηση της μεταβλητής ελέγχου
 - Μια αύξηση (ή μείωση) μέσω της οποίας η εντολή ελέγχου μεταβάλλεται κάθε φορά μέσα στο βρόχο
 - Μια συνθήκη που ελέγχει την τιμή της μεταβλητής ελέγχου κι ανάλογα να συνεχίζεται ή να τερματίζεται ο βρόχος

□ Γενική μορφή εντολής while

```
while ( συνθήκη-ελέγχου)  
    εντολή ;
```



Παράδειγμα: while

$$\sum_{i=1}^{100} i^2$$

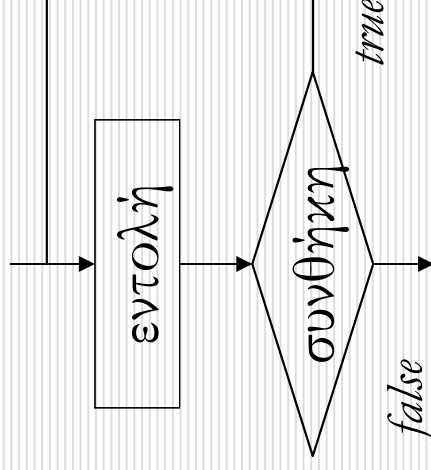
```
int sum=0, i=1;           // αρχικοποίηση
while (i<=100) {          // επανάληψη
    sum = sum + i*i;
    ++i;                  // αύξηση
}
```

Η δομή επανάληψης do - while

- ❑ Χρησιμοποιείται όταν ο αριθμός των επαναλήψεων δεν είναι γνωστός
- ❑ Ο έλεγχος στην εντολή **do-while** γίνεται στο τέλος του βρόγχου σε αντίθεση με τις εντολές for και while στις οποίες ο έλεγχος γίνεται στην αρχή του βρόγχου. Αυτό σημαίνει πως το σώμα της εντολής εκτελείται τουλάχιστον μία φορά.

❑ Γενική μορφή:

```
do {  
    εντολές;  
} while ( συνθήκη );
```



Παράδειγμα: do...while

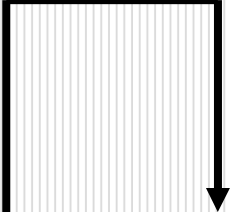
$$\sum_{i=1}^{100} i^2$$

```
int sum=0,i=1;           // αρχικοποίηση
do {
    sum = sum + i*i;
    ++i;                 // αύξηση
} while(i<=100);         // επανάληψη
```

Η εντολή break

- ❑ Προκαλεί άμεση έξοδο από μια εντολή while, for, do...while ή switch
- ❑ Η εκτέλεση του προγράμματος συνεχίζει με την αμέσως επόμενη εντολή που ακολουθεί τον κορμό μιας από τις παραπάνω εντολές

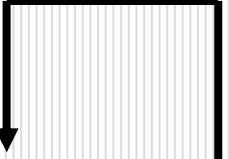
```
int sum=0,x;  
for (int i=0;i < 10; i++) {  
    scanf ("%d", &x);  
    if (x < 0)  
        break;  
    sum += x;  
}  
printf ("Sum is: %d \n", sum);
```



Η εντολή continue

- ❑ Παραλείπει τις υπόλοιπες εντολές του κορμού μιας εντολής while, for ή do...while
- ❑ Συνεχίζει με την επόμενη επανάληψη του βρόχου

```
int sum=0,x;  
for (int i=0;i < 10; i++) {  
    scanf ("%d", &x);  
    if (x < 0)  
        continue;  
    sum += x;  
}  
printf ("Sum is: %d \n", sum);
```



Ασκήσεις

□ Να γραφούν προγράμματα που να υπολογίζουν τα παρακάτω αθροίσματα.

□ 1) Αριθμητική Πρόοδος

$$\sum_{i=1}^n i = 1 + 2 + \dots + n = \frac{n(n+1)}{2}$$

$$\sum_{i=1}^n i^2 = 1^2 + 2^2 + \dots + n^2 = \frac{n(n+1)(2n+1)}{6}$$

$$\sum_{i=1}^n i^3 = 1^3 + 2^3 + \dots + n^3 = \frac{n^2(n+1)^2}{4}$$

Ασκήσεις

□ 2) Γεωμετρική Πρόοδος

$$\sum_{k=0}^n x^k = 1 + x + x^2 + \dots + x^n = \frac{x^{n+1} - 1}{x - 1}$$

□ 3) Αρμονική Πρόοδος

$$H_n = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n} = \sum_{k=1}^n \frac{1}{k}$$

□ 4) $S = 1! + 2! + 3! + \dots + N!$