



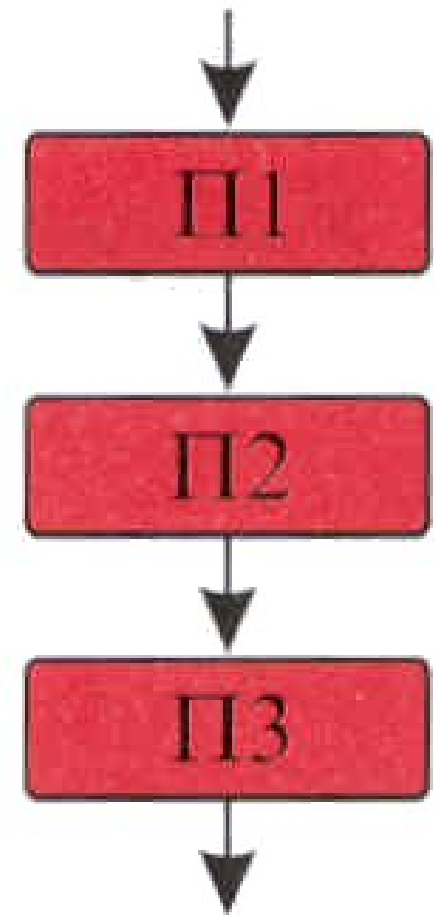
***Θεματική ενότητα 4:***  
***Έλεγχος ροής –***  
***Προτάσεις υπό συνθήκη***  
***διακλάδωσης***



## Προτάσεις ελέγχου ροής

Ο πιο συνηθισμένος τρόπος εκτέλεσης είναι ο **ακολουθιακός**:

δύο ή περισσότερες προτάσεις βρίσκονται διατεταγμένες η μία μετά την άλλη και εκτελούνται διαδοχικά.





### Προτάσεις ελέγχου ροής

- Ωστόσο, ορισμένες φορές επιβάλλεται να γίνουν λογικές επιλογές (με χρήση λογικών τελεστών και τελεστών συσχέτισης).
- Παράδειγμα:  
*ΕΑΝ* στο σηματοδότη βρίσκεται ο *ΓΡΗΓΟΡΗΣ*  
*ΤΟΤΕ* μπορείς να διασχίσεις την οδό  
*ΑΛΛΙΩΣ* περίμενε αλλαγή του σηματοδότη
- Για να επιτευχθεί οποιαδήποτε διαφοροποίηση από την ακολουθιακή εκτέλεση απαιτούνται ειδικές κατασκευές. Ορισμένες από αυτές τις κατασκευές διασφαλίζουν ταυτόχρονα τη δόμηση του προγράμματος, με κύριο στόχο: *η δομή του πηγαίου κώδικα να μας βοηθά να κατανοήσουμε τι κάνει το πρόγραμμα.* Οι κατασκευές διακρίνονται σε δύο βασικές κατηγορίες:
  - 1) την *επανάληψη* (looping)
  - 2) την *υπό συνθήκη διακλάδωση* (conditional branching)



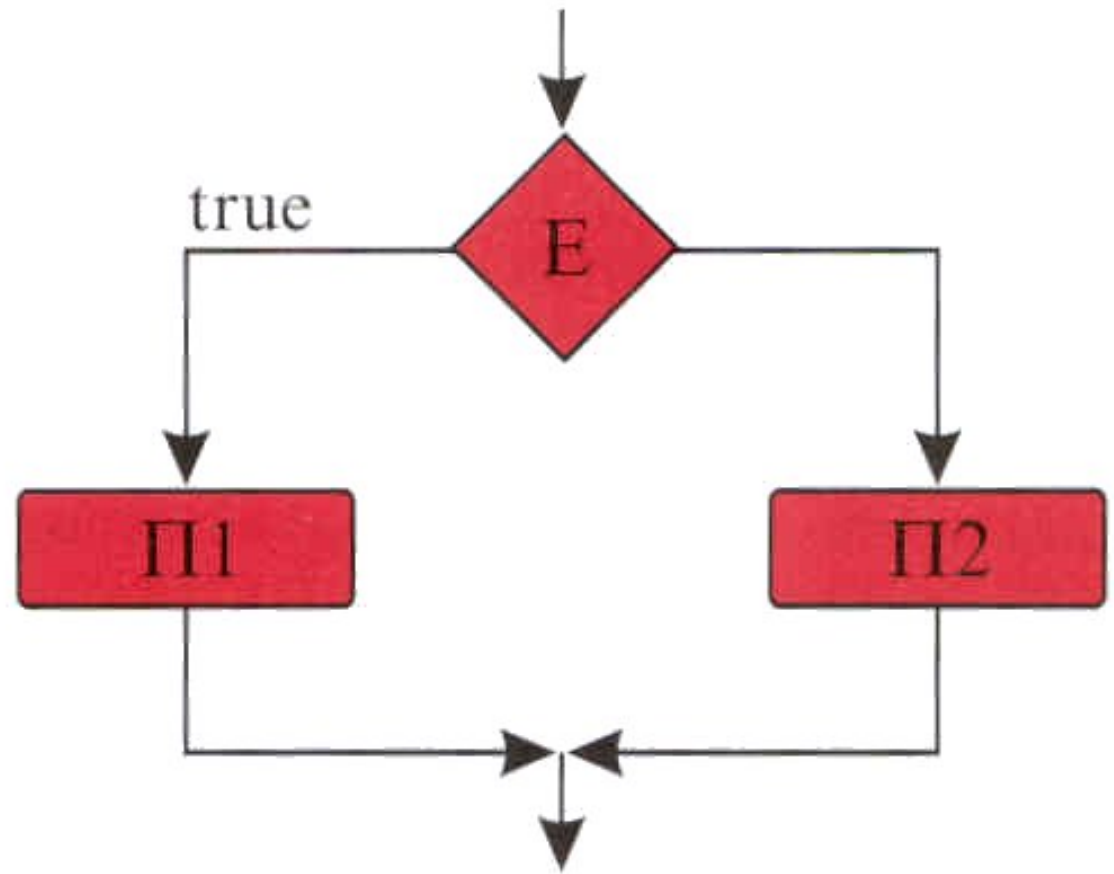
## Προτάσεις ελέγχου ροής στη C

- Προτάσεις διακλάδωσης υπό συνθήκη  
*if – else*  
*switch case*
- Προτάσεις επανάληψης  
*while*  
*do while*  
*for*
- Προτάσεις διακλάδωσης χωρίς συνθήκη  
*break*  
*continue*  
*goto*



# Επιλεκτική εκτέλεση δύο προτάσεων

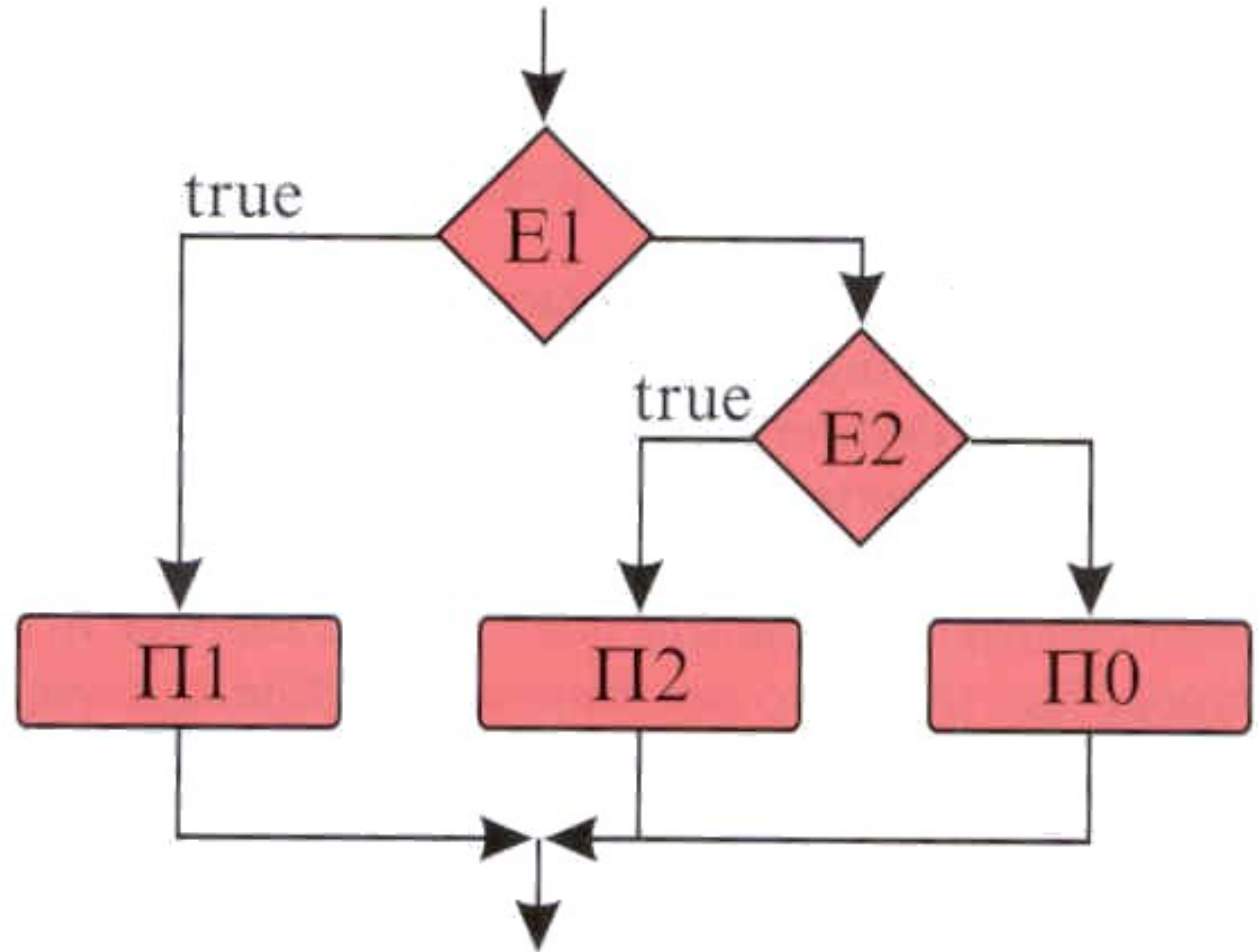
**if E1 then Π1 else Π2**





# Επιλεκτική εκτέλεση πολλών προτάσεων με ένθεση

<b>if</b>	<b>E1</b>	<b>then</b>	<b>Π1</b>
<b>else if</b>	<b>E2</b>	<b>then</b>	<b>Π2</b>
<b>else</b>			<b>Π0</b>





**Παράδειγμα:** Να περιγραφεί με ψευδοκώδικα η διεργασία που πρέπει να ακολουθήσει ο υπολογιστής για να διαπιστώσει κατά πόσο ένα δεδομένο έτος είναι δίσεκτο ή όχι.

Να χρησιμοποιηθεί η κατασκευή *if – else*.

**Λύση:**

Εάν αναπαρασταθεί το έτος με την ακέραια μεταβλητή **year** και ο **τελεστής υπολοίπου** (*modulo*) με το σύμβολο **%**, η περιγραφή μπορεί να γίνει ως ακολούθως:

*IF ((year % 400) == 0) THEN το έτος είναι δίσεκτο*

*ELSE IF ((year % 100) == 0) THEN το έτος δεν είναι δίσεκτο*

*ELSE IF ((year % 4) == 0) THEN το έτος είναι δίσεκτο*

*ELSE το έτος δεν είναι δίσεκτο*





Εναλλακτικά, ξεκινώντας από την περίπτωση το έτος να μην είναι δίσεκτο, η οποία καλύπτει την πλειοψηφία των ετών, έχουμε επιτάχυνση της εκτέλεσης του κώδικα, καθώς – εν γένει – διενεργούνται λιγότεροι έλεγχοι:

```
IF ((year % 4) != 0) THEN το έτος δεν είναι δίσεκτο  
ELSE IF ((year % 400) == 0) THEN το έτος είναι δίσεκτο  
ELSE IF ((year % 100) == 0) THEN το έτος δεν είναι δίσεκτο  
ELSE το έτος είναι δίσεκτο
```

**Πλειοψηφία των ετών**





## Υπό συνθήκη διακλάδωση *if - else*

Μία δήλωση, τρία τμήματα:

συνθήκη

```
if (light_color==green)
```

```
{
```

```
    cross_the_street();
```

```
}
```

```
else
```

```
{
```

```
    wait_for_light_to_change();
```

```
}
```

Σε απλή πρόταση τα άγκιστρα περιττεύουν



## Υπό συνθήκη διακλάδωση *if - else*

Μία δήλωση, τρία τμήματα: συνθήκη

```
if (light_color==green)
```

```
{
```

```
    cross_the_street();
```

```
}
```

```
else
```

```
{
```

```
    wait_for_light_to_change();
```

```
}
```

ΑΛΗΘΕΣ (TRUE) ΤΜΗΜΑ : μία πρόταση ή  
ένα μπλοκ προτάσεων



## Υπό συνθήκη διακλάδωση *if - else*

Μία δήλωση, τρία τμήματα: συνθήκη

```
if (light_color==green)
```

```
{
```

```
    cross_the_street();
```

```
}
```

```
else
```

```
{
```

```
    wait_for_light_to_change();
```

```
}
```

ΑΛΗΘΕΣ ΤΜΗΜΑ : μία πρόταση ή ένα μπλοκ προτάσεων

ΨΕΥΔΕΣ (FALSE) ΤΜΗΜΑ : μία πρόταση ή ένα μπλοκ προτάσεων



## Υπό συνθήκη διακλάδωση *if - else*

- Μερικές φορές δεν υπάρχει *else*, δηλαδή δεν υπάρχει ΨΕΥΔΕΣ τμήμα:

- Παράδειγμα:

```
if (gas_tank_empty == TRUE) fill_up_tank();
```

Εάν η συνθήκη είναι ψευδής (π.χ. το ντεπόζιτο της βενζίνης είναι άδειο) δε γίνεται καμία ενέργεια.



## Υπό συνθήκη διακλάδωση *if - else*

- Όταν υπάρχουν περισσότερα από δύο τμήματα και απαιτούνται ένθετες (φωλιασμένες) προτάσεις *if/else*.
- Μπορεί να αντικατασταθεί το ζεύγος

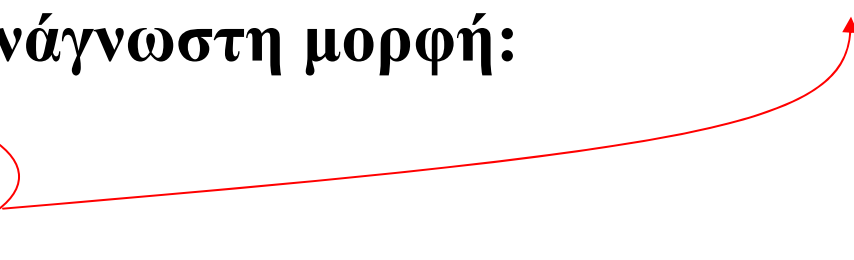
```
else {  
    if (συνθήκη) {  
        προτάσεις; }  
}
```

Π.χ.

```
if (people<5) <get a car>;  
else if (people<15) <get a van>;  
else if (people<50) <get a bus>;  
else <cancel_event>;
```

με την περισσότερο ευανάγνωστη μορφή:

```
else if (συνθήκη) {  
    προτάσεις;}
```





## Προτάσεις υπό συνθήκη διακλάδωσης

- `if (συνθήκη) πρόταση ;`

- ```
if (συνθήκη)
{
    προτάσεις;
    προτάσεις; ...
}
```

- ```
if (συνθήκη)
{
    προτάσεις;
    προτάσεις;...
}
else
{
    προτάσεις;
    προτάσεις;...
};
```

### 3 μορφές

Σημειώστε ότι η **συνθήκη** βρίσκεται πάντοτε ανάμεσα σε παρενθέσεις,

Όλα τα ΑΛΗΘΗ τμήματα και όλα τα ΨΕΥΔΗ τμήματα είναι μία μόνο πρόταση ή ένα μπλοκ προτάσεων }



**Παράδειγμα:** Να γραφεί πρόγραμμα που υπολογίζει το μέγιστο ανάμεσα σε τρεις ακέραιους

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

```
void main() {
```

```
    int a,b,c;
```

```
    scanf(“%d\n”,&a); scanf(“%d\n”,&b); scanf(“%d\n”,&c);
```

```
    if (a>b) {
```

```
        if (a>c) printf( “max(%d,%d,%d) = %d\n”,a,b,c,a );
```

```
        else printf( “max(%d,%d,%d) = %d\n”,a,b,c,c );
```

```
    }
```

```
    else if (b>c) printf( “max(%d,%d,%d) = %d\n”,a,b,c,b );
```

```
    else printf( “max(%d,%d,%d) = %d\n”,a,b,c,c );
```

```
}
```



# Υποθετικός τελεστής

Ο υποθετικός τελεστής (`?:`) αποτελείται από δύο σύμβολα. Ανήκει στην κατηγορία των τελεστών που αποτελούνται από συνδυασμό συμβόλων και δεν ακολουθούν καμία από τις postfix, prefix ή infix σημειογραφίες. Όταν τα σύμβολα ή οι λέξεις του τελεστή είναι διάσπαρτα στους τελεστέους στους οποίους εφαρμόζεται ο τελεστής, λέμε ότι ο τελεστής είναι σε **μεικτή σημειογραφία** (mixfix notation).





## Υποθετικός τελεστής (συνέχεια)

- Η έκφραση που σχηματίζει ο υποθετικός τελεστής έχει τη μορφή:

$\text{εκφρ1} \text{ ? εκφρ2 : εκφρ3}$

- Η τιμή της παραπάνω έκφρασης είναι η τιμή της **εκφρ2**, εάν η **εκφρ1** είναι αληθής, αλλιώς είναι η τιμή της **εκφρ3**.

- Η **εκφρ1** αποτελεί τη συνθήκη ελέγχου. Έτσι η έκφραση

$x > z \text{ ? } x \text{ : } z$

- έχει τιμή **x**, εάν το **x > z** είναι αληθές, διαφορετικά έχει τιμή **z**.



**Παράδειγμα:** Να γραφεί κώδικας που βρίσκει το μεγαλύτερο δύο και τριών ακεραίων αριθμών και τον τυπώνει.

**Δύο αριθμοί:**

```
printf( "maximum is %d\n", (num1>num2)?num1:num2 );
```

**Τρεις αριθμοί:**

```
max=(num1>num2?num1:num2) > num3 ? (num1>num2?num1:num2):num3;  
printf( "maximum is %d\n", max );
```

*Προκύπτει ένας αριθμός και συγκρίνεται με τον **num3**. Εάν ισχύει η ανισότητα αυτός ο αριθμός θα τυπωθεί ως μέγιστος, αλλιώς ο **num3**.*



## Υπό συνθήκη διακλάδωση *switch()*

```
switch(έκφραση)
{
  case (σταθ.-έκφρ. 1):
    προτάσεις;
    break;
  case (σταθ.-έκφρ. 2):
    προτάσεις;
    break;
  default:
    προτάσεις;
    break;
}
```

•Όταν υπάρχουν πολλές ένθετες προτάσεις *if* ή *elseif*, υπάρχει δυσκολία στην ανάγνωση και καθυστέρηση στην εκτέλεση.

•Τότε προτιμάται η χρήση της *switch()*.



### *switch()* (συνέχεια)

- Η πρόταση *switch* επιτρέπει τον προσδιορισμό απεριόριστου αριθμού διαδρομών ανάλογα με την τιμή της έκφρασης.
- Υπολογίζεται η έκφραση και η τιμή της συγκρίνεται διαδοχικά με τις σταθερές εκφράσεις (*σταθ.-έκφρ. 1, σταθ.-έκφρ. 2, ...*). Ο έλεγχος μεταφέρεται στις προτάσεις που είναι κάτω από τη σταθερά έκφραση με την οποία ισούται η τιμή της *έκφρασης*.
- Εάν δεν ισούται με καμία από τις σταθερές εκφράσεις, ο έλεγχος μεταφέρεται στις προτάσεις που ακολουθούν την ετικέτα *default*, εάν αυτή υπάρχει, αλλιώς στην πρόταση που ακολουθεί το σώμα της *switch*.



### *switch()* (συνέχεια)

- Η πρόταση ελέγχου *break*, η οποία υποδηλώνει άμεση έξοδο από τη *switch*, είναι προαιρετική. Εάν αυτή λείπει, η εκτέλεση των προτάσεων που ακολουθούν την επιλεγείσα ετικέτα θα ακολουθήσει από την εκτέλεση και των προτάσεων και των επόμενων *case* ετικετών.
- Στην πράξη η *break* συναντάται σχεδόν πάντοτε, ακόμη και μετά τις προτάσεις τής *default*. Το τελευταίο γίνεται για να προστατευθούμε από το δύσκολο στην ανεύρεση σφάλμα που θα προκύψει από μελλοντική προσθήκη μίας νέας ετικέτας με ταυτόχρονη παράλειψη προσθήκης πριν από αυτή τής *break*.



### *switch()* (συνέχεια)

Η λειτουργία της *switch* διέπεται από το ακόλουθο σύνολο κανόνων:

- Κάθε *case* πρέπει να έχει μία *int* ή *char* σταθερά έκφραση.
- Δύο *case* δεν μπορούν να έχουν την ίδια τιμή.
- Οι προτάσεις κάτω από την ετικέτα *default* εκτελούνται όταν δεν ικανοποιείται καμία από τις *case* ετικέτες.
- Η *default* δεν είναι απαραίτητα η τελευταία ετικέτα.
- Οι *case* και *default* μπορούν να τοποθετηθούν με οποιαδήποτε σειρά.
- Η *break* μετά την τελευταία ετικέτα αποτελεί καλή τακτική, αν και δεν είναι απαραίτητη.



**Παράδειγμα:** Να γραφεί τμήμα κώδικα, το οποίο να δίνει τη δυνατότητα στο χρήστη να εισάγει δύο αριθμούς και στη συνέχεια να εκτελεί επί αυτών επιλεκτικά μία από τις τέσσερις αριθμητικές πράξεις

### **Λύση:**

Χρησιμοποιώντας “δομημένα Ελληνικά”, η διεργασία περιγράφεται ως εξής:

- 1. πάρε δύο αριθμούς*
- 2. ενημέρωσε το χρήστη για δυνατές επιλογές*
- 3. πάρε την επιλογή του χρήστη*
- 4. ανάλογα με την επιλογή*
- 5. εκτέλεσε την αντίστοιχη πράξη*
- 6. εμφάνισε το αποτέλεσμα*
- 7. τερμάτισε*

Ο κώδικας του προγράμματος είναι ο ακόλουθος:



# Προγραμματισμός Ι

```
#include <stdio.h>      #include <math.h>
#define MY_ZERO 1E-5
#define ADD 1
#define SUB 2
#define MUL 3
#define DIV 4

main(){
    float num1, num2,result;
    int choice, flag=0;
    printf( "\nGive first number:"); scanf("%f",&num1 );
    printf( "\nGive second number:"); scanf("%f",&num2 );
    printf( "\n Select one of the following:" );
    printf( "\n\t\t\t %d -> + (addition)\n",ADD );
    printf( "\n\t\t\t %d -> - (subtraction)\n",SUB );
    printf( "\n\t\t\t %d -> * (multiplication)\n",MUL );
    printf( "\n\t\t\t %d -> / (division)\n",DIV );
    scanf( "%d",&choice );
```

Κάθε μία από τις πράξεις  
αντιστοιχίζεται σε έναν ακέραιο





```
switch(choice){
  case ADD:
    result=num1+num2;
    break;
  case SUB:
    result=num1-num2;
    break;
  case MUL:
    result=num1*num2;
    break;
  case DIV:
    if (fabs(num2)>MY_ZERO) result=num1/num2; // num2 != 0
    else { printf( "\\t\\t ERROR: division by 0" ); flag=1; }
    break;
  default:
    printf( "This selection is not supported" );
    break;
} // τέλος της switch
if (!flag) printf( "\\n\\tResult: %f\\n",result ); } // τέλος της main
```



## Αποτελέσματα:

```
C:\temp\try.exe

Give first number:-1.23
Give second number:0.0000002
Select one of the following:
    1 -> + (addition)
    2 -> - (subtraction)
    3 -> * (multiplication)
    4 -> / (division)
4
ERROR: division by 0
```

```
C:\temp\try.exe

Give first number:-1.23
Give second number:0.666
Select one of the following:
    1 -> + (addition)
    2 -> - (subtraction)
    3 -> * (multiplication)
    4 -> / (division)
4
Result: -1.846847
```