

Η γλώσσα προγραμματισμού C



Εντολές ελέγχου ροής προγράμματος
(if-else & switch)

Κ. Βασιλάκης, ΣΤΕΦ, ΤΕΙ Κρήτης



Η εντολή if-else

- Η εντολή if-else υπάρχει σχεδόν σε όλες τις γλώσσες προγραμματισμού.
- Χρησιμοποιείται για τον έλεγχο της ροής εκτέλεσης του προγράμματος, εξετάζοντας αν ισχύει ή όχι κάποια **συνθήκη**.
- Η **συνθήκη** μπορεί να είναι μια λογική έκφραση/πρόταση, μια έκφραση συσχετισμού, ένα αποτέλεσμα κάποιας πράξης, είτε ακόμα και κάποια μεταβλητή ή τιμή.
- Όταν η συνθήκη είναι αληθής τότε επιλέγεται να εκτελεστεί μια ενότητα (ομάδα-block) εντολών. Διαφορετικά (else) επιλέγεται μια άλλη ενότητα εντολών.
- Αποτελεί βασική δομή προγραμματισμού με την οποία αναπαριστούμε στα προγράμματα μας λήψεις αποφάσεων ανάλογα με τις συνθήκες που επικρατούν.



Η σύναξη της εντολής if-else

- Στην C η γενική σύνταξη της εντολής είναι:

if (<ΣΥΝΘΗΚΗ>)

Ενότητα_Εντολών_A

else

Ενότητα_Εντολών_B

- Οι **Ενόητες Εντολών** μπορεί να περιλαμβάνουν μία εντολή ή πολλές εντολές (block) που περικλείονται σε άγκιστρα (**{,}**).
- Οι εντολές (ή η εντολή) της **Ενόητας_Εντολών_A** εκτελούνται αν η <ΣΥΝΘΗΚΗ> είναι αληθής, ενώ οι εντολές (ή η εντολή) της **Ενόητας_Εντολών_B** εκτελούνται αν η <ΣΥΝΘΗΚΗ> είναι ψευδής.
- <ΣΥΝΘΗΚΗ> αληθής: διάφορη του μηδενός ($\neq 0$)
- <ΣΥΝΘΗΚΗ> ψευδής: ίση με μηδέν ($= 0$)



Παρατηρήσεις

- Αν η ενότητα εντολών περιέχει μόνο μια εντολή, τότε τα άγκιστρα μπορούν να παραλειφθούν.

if (a<b) max = b;

- Δεν βάζουμε το ελληνικό ερωτηματικό «;» στο τέλος της **if** ή της **else** εντολής. Το ερωτηματικό ουσιαστικά τερματίζει στο σημείο εκείνο την εντολή.
- Το **else** είναι προαιρετικό και όταν υφίσταται αναφέρεται στο πλησιέστερο προηγούμενο **if** που δεν έχει **else**.
- Αν δεν υπάρχει **else** και η συνθήκη είναι ψευδής, τότε η εντολή **if** δεν κάνει τίποτα και το πρόγραμμα προχωρά στην εκτέλεση των επόμενων εντολών.
- Κάθε ενότητα είναι δυνατόν να περικλείει και άλλες **if-else** ενότητες (blocks) εντολών (ένθετα – nested if).
- Προσοχή ο τελεστής ελέγχου ισότητας == (διπλό ίσον) είναι διαφορετικός από τον τελεστή εκχώρησης = (μονό ίσον).



Γραφική αναπαράσταση της πλήρους δομής

Πριν τη δομή το πρόγραμμα έχει εκτελέσει τις προηγούμενες εντολές

(έναρξη)
Εντολές
Προγράμματος
.....

Γίνεται εκτίμηση της συνθήκης

Συνθήκη $\neq 0$

ΑΛΗΘΗΣ

ΣΥΝΘΗΚΗ

ΨΕΥΔΗΣ

Συνθήκη = 0

Εκτελούνται
αν η ΣΥΝΘΗΚΗ
είναι αληθής

Ενότητα
Εντολών A

Ενότητα
Εντολών B

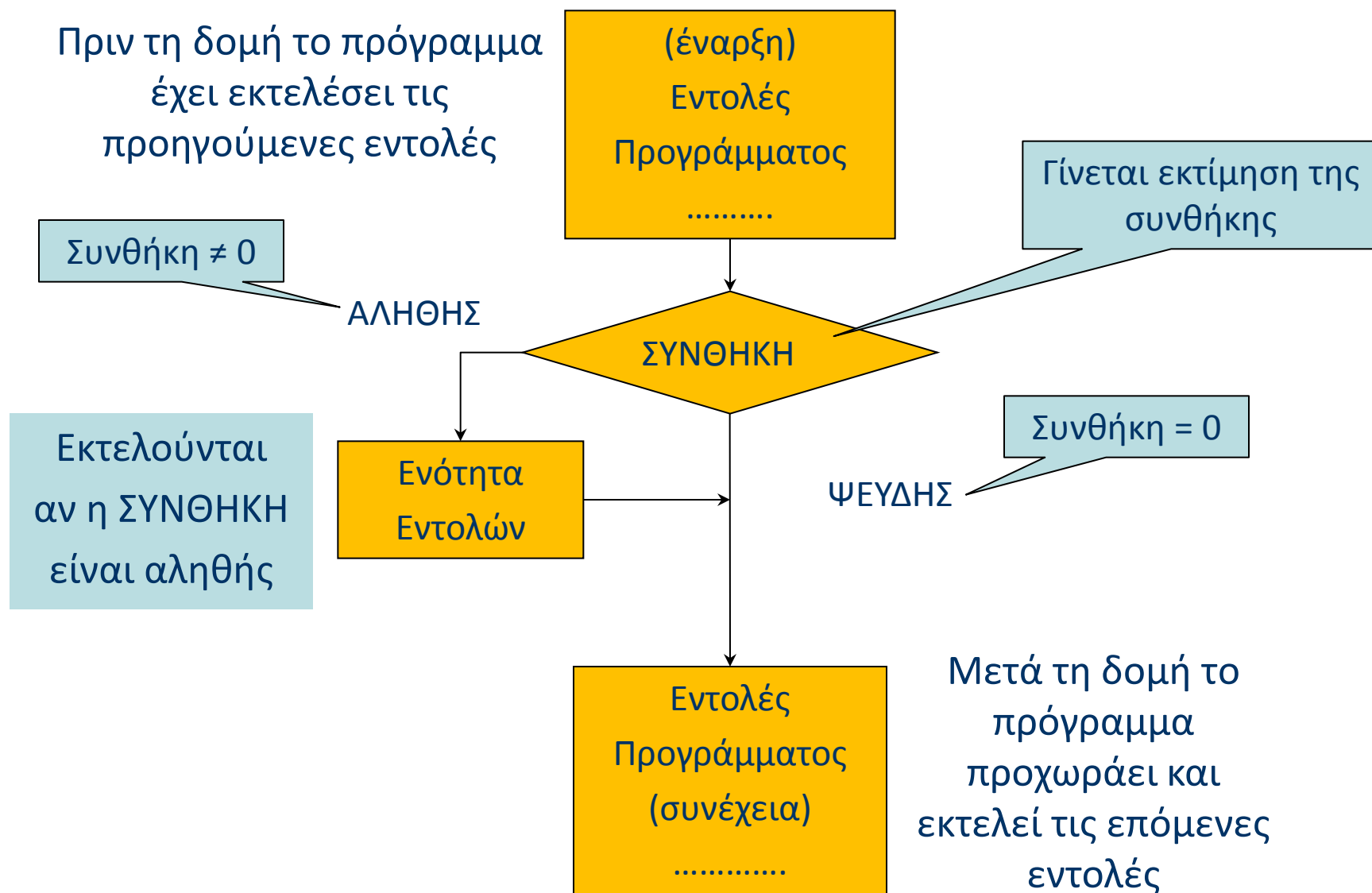
Εκτελούνται
αν η ΣΥΝΘΗΚΗ
είναι ψευδής

Εντολές
Προγράμματος
(συνέχεια)
.....

Μετά τη δομή το πρόγραμμα προχωράει και εκτελεί τις επόμενες εντολές



Γραφική αναπαράσταση της δομής χωρίς το else



Παραδείγματα της δομής if

```
printf("Δώστε έναν ακέραιο:");  
scanf("%d", &a);  
if (a == 0)  
{  
    printf ("ο ακέραιος %d είναι μηδέν\n",a);  
    a++;  
}
```

```
if (a == 0)  
{  
    printf ("ο ακέραιος %d είναι μηδέν\n",a);  
    a++;  
}  
else printf ("ο ακέραιος %d είναι δεν είναι μηδέν\n",a);
```

```
if (a<=0) a++;
```

```
if (a<=0) a++;  
else a--;
```

```
if (a<=0) {  
    a=a+b;  
    b--;  
}  
else {  
    a=a-b;  
    b++;  
}
```



Παραδείγματα προγραμμάτων με την if-else

```
#include <stdio.h>
main()
{
    int a,b,min,max;
    printf("Δώστε a:");
    scanf("%d", &a);
    printf("Δώστε b:");
    scanf("%d", &b);
    if (a<b) min = a;
    else min = b;
    if (a<b) max = b;
    else max = a;
    printf("min = %d max = %d\n", min, max);
}
```

Ενότητες με μία εντολή (δεν χρειάζονται αγκιστρα)

```
int a,b;
printf("Δώστε a:");
scanf("%d", &a);
printf("Δώστε b:");
scanf("%d", &b);
if (a <= b)
{
    a = a+1;
    b = b/3;
}
else
{
    a = a-1;    b = b%3;
}
printf("a= %d, b= %d \n", a,b);
```

Ενότητες με πολλές εντολές (με άγκιστρα)



Ένθετα (nested) if

```
if (num == 2)
    printf("The num is now: 2\n");
if (num < 5)
    printf("The num is now %d, less than 5\n",num);
else
    printf("The num is now %d, greater than 4\n",num);
```

Έλεγχος της μεταβλητής num

σε τι διαφέρουν?

```
if (num == 2)
    printf("The num is now: 2\n");
else if (num < 5)
    printf("the num is now less than 5\n");
else
    printf("the num is now greater than 4\n");
```

Ένθετο if



Εύρεση μεγαλύτερου από 3 αριθμούς

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

```
.....
```

Με ένθετα if

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

```
    if (a>c)
```

```
        printf ("max is a:%d", a);
```

```
    else
```

```
        printf ("max is c:%d", c);
```

```
else
```

```
    if (b>c)
```

```
        printf ("max is b:%d",b);
```

```
    else
```

```
        printf("max is c:%d",c);
```

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

```
.....
```

Με λογικές προτάσεις

```
if (a>b && a>c)
```

```
    printf ("max is a:%d", a);
```

```
if (b>a && b > c)
```

```
    printf ("max is b:%d", b);
```

```
if (c> a && c > b)
```

```
    printf ("max is c:%d",c);
```

Πως συμπεριφέρονται αυτά τα προγράμματα αν
 $a=b$ ή $b=c$;



Αριθμομηχανή

```
char p;  
float x, y, result;  
int flag=1;  
scanf("%f %c %f", &x, &p, &y);  
if (p == '+') result = x + y;  
else if (p == '-') result = x - y;  
    else if (p == '*') result = x * y;  
        else if (p == '/') result = x / y;  
            else flag=0;  
if (flag !=0) printf("%f %c %f = %f\n", x, p, y, result);  
else printf("Λάθος τελεστής\n");
```

Λείπει ένα printf που
θα εξηγεί τι πρέπει να
εισάγει ο χρήστης

Τι επέμβαση
απαιτείται αν
η y είναι ίση
με μηδέν;



Περίεργες συνθήκες

- Η έκφραση

if (1) είναι πάντα αληθής

- Η έκφραση

if (0) είναι πάντα ψευδής

- Η εκφραση

if (a) είναι ισοδύναμη με την *if (a != 0)*

- Η εκφραση

if (!a) είναι ισοδύναμη με την *if (a == 0)*

- Η εκφραση

if (a=-2) είναι αληθής

Γενικά, η έκφραση $a = \langle \text{τιμή} \rangle$ είναι αληθής αν η τιμή είναι $\neq 0$. Αν η τιμή που αναθέτουμε είναι 0 τότε η συνθήκη είναι ψευδής.



Η εντολή switch

- Η γενικευμένη μορφή της εντολής είναι η εξής:

switch (έκφραση)

{

case σταθερά_1:

Ενότητα εντολών -1;

εκτελούνται αν η σταθερά_1=έκφραση

break;

case σταθερά_2:

Ενότητα εντολών _2 ;

εκτελούνται αν η σταθερά_2=έκφραση

break;

.....

default:

Ενότητα εντολών ;

εκτελούνται αν η έκφραση δεν είναι ίση με καμιά σταθερά των cases

break;

}

Εντολή που χρησιμοποιείται αντί της if-else



Αριθμομηχανή με switch

```
char p;  
float x, y, result;  
int flag=0;  
scanf("%f %c %f", &x, &p, &y);  
switch(p) {  
    case '+': result = x + y; break;  
    case '-': result = x - y; break;  
    case '*': result = x * y; break;  
    case '/': result = x / y; break;  
    default: printf("Λάθος τελεστής\n");  
            flag=1;  
}  
if (flag==0)  
    printf("%f %c %f = %f\n", x, p, y, result);
```

Τι επέμβαση
απαιτείται αν
η y είναι ίση
με μηδέν;



Ανάλυση της δομής switch

- Εκτελούνται οι εντολές των οποίων η σταθερά του **case** ταιριάζει με την τιμή της έκφρασης.
- Η τιμή της έκφρασης στη **switch** και οι τιμές των σταθερών των **cases** πρέπει να είναι ακέραιοι ή χαρακτήρες.
- Η τιμή της έκφρασης συγκρίνεται διαδοχικά με τις σταθερές των **cases**.
- Αν η τιμή της έκφρασης ταιριάζει με τη σταθερά ενός **case** τότε εκτελούνται οι αντίστοιχες εντολές του **case**.
- Αν η τιμή της έκφρασης δεν ταιριάζει με καμία σταθερά, τότε εκτελείται η εντολή στο block της default (που είναι προαιρετικό).
- Η εκτέλεση του προγράμματος μέσα σε ένα switch συνεχίζεται στο επόμενο case, εκτός αν δεν μεσολαβεί κάποια από τις εντολές **break**, **exit** ή **return**.



Παρατηρήσεις επί της δομής switch

- Δεν χρησιμοποιούμε «;» στο τέλος των case και default, αλλά το «:».
- Όταν δεν υπάρχει **default** και η τιμή της έκφρασης δεν είναι ίση με τις σταθερές των case, τότε τερματίζεται η εντολή switch και εκτελούνται οι εντολές που ακολουθούν τη switch.
- Η **default** δεν μπαίνει απαραίτητα στο τέλος της switch.
- Προφανώς **δεν μπορούν** να έχουν **την ίδια τιμή δύο σταθερές** των cases.
- Η εντολή **break** προκαλεί τον τερματισμό μιας εντολής switch (έξοδο) ή και μιας επαναληπτικής διαδικασίας (που θα δούμε στα επόμενα μαθήματα).
- Με την εντολή **break** εκτέλεση του προγράμματος συνεχίζεται μετά την εντολή switch ή την εντολή επανάληψης.



Αριθμομηχανή με switch – άλλος τρόπος

```
switch(p) {  
    case '+':  
        result = x + y; printf("%f %c %f = %f\n", x, p, y, result);  
        break;  
    case '-':  
        result = x - y; printf("%f %c %f = %f\n", x, p, y, result);  
        break;  
    case '*':  
        result = x * y; printf("%f %c %f = %f\n", x, p, y, result);  
        break;  
    case '/':  
        if (y == 0) printf("Διαίρεση με 0 \n");  
        else printf("%f %c %f = %f\n", x, p, y, result);  
        break;  
    default:  
        printf("Λάθος τελεστής\n");  
}
```



Στο εργαστήριο

- Μεταγλωττίστε τα προγράμματα που υπάρχουν στο περιεχόμενο της θεωρίας του 4^{ου} μαθήματος.
- Γράψτε ένα πρόγραμμα στο οποίο θα δίνουμε σαν δεδομένο έναν ακέραιο και αυτό θα αποφαινεται αν αυτός είναι άρτιος ή περιττός.
- Γράψτε ένα πρόγραμμα το οποίο θα συσχετίζει 2 αριθμούς που θα δίδονται από το πληκτρολόγιο (δηλαδή αν είναι ίσοι ή ποιός είναι μεγαλύτερος από τον άλλο).
- Γράψτε ένα πρόγραμμα το οποίο θα υπολογίζει και θα εμφανίζει στην οθόνη τις πραγματικές ρίζες μιας δευτεροβάθμιας εξίσωσης. Επίσης, θα διακρίνει αν υπάρχουν 2 ρίζες ή 1 δίνοντας τα αντίστοιχα μηνύματα. Τέλος, αν οι ρίζες είναι μιγαδικές θα βγαίνει το κατάλληλο μήνυμα.
- Να γραφεί πρόγραμμα σε γλώσσα C όπου θα διαβάζεται ένας αριθμός από 1 μέχρι 7 (να γίνεται έλεγχος τιμής) και να εμφανίζεται η αντίστοιχη ημέρα της εβδομάδας (1 Κυριακή, 2 Δευτέρα, 3 Τρίτη, κλπ). Μετατρέψτε το πρόγραμμα ώστε με δεδομένα 1 έως 12 να τυπώνεται ο αντίστοιχος μήνας.

