

ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ

Εντολή if

1

Τι χρειάζεται η εντολή if ;

Η εντολή **if** επιτρέπει την επιλεκτική εκτέλεση εντολών ελέγχοντας μια συνθήκη.

2

Παράδειγμα #1

Κατασκευάστε πρόγραμμα που θα βρίσκει το μεγαλύτερο από δύο ακέραιους a, b που θα εισάγονται από το πληκτρολόγιο.

3

Παράδειγμα #1

```
#include <iostream>
using namespace std;

int main ( )
{
    int a, b;                // Οι δύο αριθμοί
    int max;                // Ο μεγαλύτερος

    cout << "Εισάγετε δύο ακέραιους ";
    cin >> a >> b;

    if ( a > b )
        max = a;
    else
        max = b;

    cout << "Ο μεγαλύτερος είναι ο " << max << endl;
}
```

4

Συντακτικό της εντολής if 1/4

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

Παρατηρήσεις:

- Οι εντολές γράφονται μερικά κενά (ή ένα tab) πιο δεξιά για ευκρίνεια.
- Δεν βάζουμε ερωτηματικό μετά τη σύγκριση ούτε μετά την else.
- Το τμήμα else μπορεί να παραληφθεί οπότε η εντολή γίνεται:

```
if ( σύγκριση )
    εντολή ;
```

5

Τι συγκρίσεις μπορεί να γίνουν ;

Μεταξύ δύο μεταβλητών ή γενικότερα δύο παραστάσεων *a* και *b* μπορούν να γίνουν οι παρακάτω συγκρίσεις:

Σύγκριση	Τι σημαίνει
$a == b$	$a = b$
$a != b$	$a \neq b$
$a > b$	$a > b$
$a < b$	$a < b$
$a >= b$	$a \geq b$
$a <= b$	$a \leq b$

Κάθε σύγκριση είναι είτε **αληθής** είτε **ψευδής**.
Στη C++ το αληθές είναι ο ακέραιος 1 ενώ το ψευδές ο ακέραιος 0

6

Πως λειτουργεί η εντολή if

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

Πρώτα γίνεται η **σύγκριση**

Εάν η **σύγκριση** είναι **αληθής** εκτελείται η *εντολή1* και το πρόγραμμα συνεχίζει μετά την *εντολή2*.

Εάν η **σύγκριση** είναι **ψευδής** εκτελείται η *εντολή2*.

7

Συντακτικό της εντολής if 2/4

```
if ( σύγκριση ) {
    εντολή ;
    εντολή ;
    :
} else {
    εντολή ;
    εντολή ;
    :
}
```

Σε περίπτωση που θέλουμε να εκτελεστούν **περισσότερες από μια εντολές** είτε όταν η σύγκριση είναι αληθής είτε όταν είναι ψευδής, χρησιμοποιούμε άγκιστρα για να ομαδοποιήσουμε τις εντολές.

8

Συντακτικό της εντολής if 3/4

Σημαντική παρατήρηση:

Η εντολή if από το σημείο που αρχίζει μέχρι το σημείο που τελειώνει (συμπεριλαμβανομένων και όλων των εντολών που περιέχει)

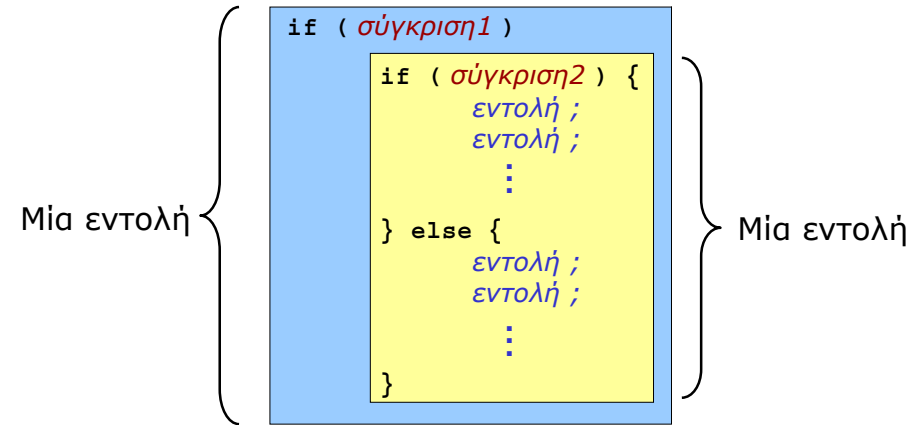
θεωρείται συντακτικά ως μία εντολή.

Αυτό έχει σημασία στα σημεία όπου το συντακτικό της γλώσσας απαιτεί **μία εντολή**.

9

Συντακτικό της εντολής if 4/4

Παράδειγμα: Δύο ένθετες εντολές if



10

Παράδειγμα #2

Κατασκευάστε πρόγραμμα που θα επιλύει την εξίσωση πρώτου βαθμού

$$Ax+B=0$$

όταν δίνονται τα A και B.

Υπενθύμιση: Η λύση είναι $-\frac{B}{A}$ αν $A \neq 0$

αλλιώς δεν υπάρχει λύση.

11

Παράδειγμα #2

```

#include <iostream>
using namespace std;

int main ( )
{
    double a, b;           // Οι συντελεστές
    double x;              // Η λύση

    cout << "Εισάγετε τους συντελεστές \n";
    cin >> a >> b;

    if ( a != 0 ) {
        x = -b/a;
        cout << "Η λύση είναι " << x << endl;
    } else
        cout << "Δεν υπάρχει λύση \n";
}

```

12

Παράδειγμα #3

Κατασκευάστε πρόγραμμα που θα βρίσκει τις πραγματικές λύσεις (αν υπάρχουν) της δευτεροβάθμιας εξίσωσης

$$Ax^2 + Bx + C = 0$$

όταν δίνονται τα A, B και C.

Υπενθύμιση:

- Αν A=0 τότε έχουμε μια εξίσωση πρώτου βαθμού.
- Οι λύσεις της δευτεροβάθμιας είναι:

$$x_{1,2} = \frac{-B \pm \sqrt{B^2 - 4AC}}{2A}$$

13

Παράδειγμα #3

```
#include <iostream>
#include <cmath>
using namespace std;

int main ( )
{
    double a, b, c;           // Οι συντελεστές
    double d;                 // Η διακρίνουσα
    double x1, x2;           // Οι λύσεις

    cout << "Εισάγετε τα a,b,c ";
    cin >> a >> b >> c;

Συνεχίζεται...
```

14

Παράδειγμα #3

```
...Συνέχεια
if ( a != 0 ) {           // Είναι εξίσωση 2ου βαθμού
    d = b*b-4*a*c;
    if ( d > 0 ) {
        x1 = (-b+sqrt(d))/(2*a);
        x2 = (-b-sqrt(d))/(2*a);
        cout << "Υπάρχουν δύο λύσεις:\n";
        cout << x1 << endl;
        cout << x2 << endl;
    } else if ( d == 0 ) {
        x1 = -b/(2*a);
        cout << "Υπάρχει μία λύση:\n";
        cout << x1 << endl;
    } else
        cout << "Δεν υπάρχουν λύσεις\n";
Συνεχίζεται...
```

15

Παράδειγμα #3

```
...Συνέχεια
    } else               // Είναι εξίσωση 1ου βαθμού
        if ( b != 0 ) {
            x1 = -c/b;
            cout << "Υπάρχει μια λύση:\n";
            cout << x1 << endl;
        } else
            cout << "Δεν υπάρχει λύση\n";
}
```

16

Σύνθετες λογικές παραστάσεις

Μπορούμε να κατασκευάσουμε σύνθετες λογικές παραστάσεις με τη χρήση των **λογικών τελεστών**

&& **||** **!**

`if (σύγκριση1 && σύγκριση2)`

`if (σύγκριση1 || σύγκριση2)`

`if (! σύγκριση)`

Το αποτέλεσμα μιας σύνθετης λογικής παράστασης είναι είτε αληθές είτε ψευδές και εξαρτάται από τις επιμέρους συγκρίσεις.

17

Πίνακες αλήθειας (&&)

σύγκριση1	σύγκριση2	σύγκριση1 && σύγκριση2
A	A	A
A	Ψ	Ψ
Ψ	A	Ψ
Ψ	Ψ	Ψ

Πρακτικός κανόνας:

Το τελικό αποτέλεσμα είναι αληθές όταν
και οι δύο
συγκρίσεις είναι αληθείς

18

Πίνακες αλήθειας (||)

σύγκριση1	σύγκριση2	σύγκριση1 σύγκριση2
A	A	A
A	Ψ	A
Ψ	A	A
Ψ	Ψ	Ψ

Πρακτικός κανόνας:

Το τελικό αποτέλεσμα είναι αληθές όταν
είτε η μία είτε η άλλη
σύγκριση είναι αληθείς

19

Πίνακες αλήθειας (!)

σύγκριση	! σύγκριση
A	Ψ
Ψ	A

Ο τελεστής **!** **αντιστρέφει** το αποτέλεσμα της σύγκρισης.

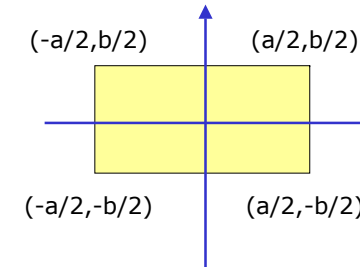
20

Παράδειγμα #4

Ένα παραλληλόγραμμο με κέντρο την αρχή των αξόνων έχει πλευρές a , b . Βρείτε αν ένα σημείο με συντεταγμένες x , y βρίσκεται εντός του παραλληλογράμμου.

21

Παράδειγμα #4



Υπενθύμιση:

- Η συντεταγμένη x θα πρέπει να βρίσκεται μεταξύ $-a/2$ και $a/2$
- Όμοια η συντεταγμένη y θα πρέπει να βρίσκεται μεταξύ $-b/2$ και $b/2$

22

Παράδειγμα #4

```
#include <iostream>
using namespace std;

int main ( )
{
    double a, b;           // Τα μήκη των πλευρών
    double x, y;           // Συντεταγμένες σημείου
    double a2, b2;         // Το ήμισυ των πλευρών

    cout << "Εισάγετε τα μήκη των πλευρών ";
    cin >> a >> b;

    cout << "Εισάγετε τις συντεταγμένες ";
    cin >> x >> y;
```

Συνεχίζεται...

23

Παράδειγμα #4

```

...Συνέχεια

    a2 = a/2;
    b2 = b/2;

    if ( x>=-a2 && x<=a2 && y>=-b2 && y<=b2 )
        cout << "Το σημείο είναι εντός\n";
    else
        cout << "Το σημείο είναι εκτός\n";
}
```

24

Προτεραιότητες λογικών τελεστών

Σε μια σύνθετη λογική παράσταση όπως πχ.

```
a>0 && b>0 || c!=2
```

η σειρά των λογικών πράξεων καθορίζεται από την προτεραιότητα των τελεστών. Για τελεστές ίδιας προτεραιότητας οι πράξεις γίνονται από αριστερά προς τα δεξιά.

Τελεστής Προτεραιότητα

!	Υψηλή
&&	
	Χαμηλή

Εάν υπάρχουν πράξεις σε παρενθέσεις, αυτές γίνονται πάντα πρώτες.

Προτεραιότητες

Τελεστής	Προτεραιότητα	Προσεταιριστικότητα
! ++ --	Υψηλή	Από δεξιά προς αριστερά
* / %		Από αριστερά προς δεξιά
+ -		Από αριστερά προς δεξιά
< <= > >=		Από αριστερά προς δεξιά
== !=		Από αριστερά προς δεξιά
&&		Από αριστερά προς δεξιά
		Από αριστερά προς δεξιά
= += -= *= /= %=	Χαμηλή	Από δεξιά προς αριστερά

Παράδειγμα #5

Με ποια εντολή if μπορούμε να διαπιστώσουμε αν ένας ακέραιος είναι άρτιος ή περιττός ;

Υπενθύμιση:

Ένας ακέραιος είναι άρτιος αν είναι πολλαπλάσιο του δύο, δηλαδή διαιρείται ακριβώς με το δύο χωρίς να αφήνει υπόλοιπο.

Παράδειγμα #5

```
if ( k%2 == 0 )
    ... είναι άρτιος ...
else
    ... είναι περιττός ...
```

Σημείωση:

Ο τελεστής % έχει μεγαλύτερη προτεραιότητα από τον τελεστή ==

Παράδειγμα #6

Κατασκευάστε πρόγραμμα που θα αποφασίζει αν ένα έτος είναι δίσεκτο ή όχι.

Υπενθύμιση:

Ένα έτος θεωρείται δίσεκτο όταν διαιρείται ακριβώς με το 4. Όμως από αυτά εξαιρούνται τα έτη που διαιρούνται με το 100, εκτός αν διαιρούνται με το 400.

29

Παράδειγμα #6

```
#include <iostream>
using namespace std;

int main ( )
{
    int year;

    cout << "Εισάγετε το έτος\n";
    cin >> year;

    if ( year%4 == 0 )
        if ( year%100 == 0 )
            if ( year%400 == 0 )
                cout << "Είναι δίσεκτο\n";
            else
                cout << "Δεν είναι δίσεκτο\n";
        else
            cout << " Είναι δίσεκτο \n";
    else
        cout << " Δεν είναι δίσεκτο \n";
}
```

30

Παράδειγμα #6

Εναλλακτικά θα μπορούσαμε να γράψουμε:

```
if (year%4 == 0 && year%100 != 0 || year%400 == 0)
    ... είναι δίσεκτο ...
else
    ... δεν είναι δίσεκτο ...
```

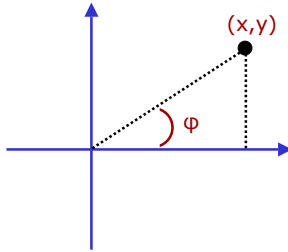
31

Παράδειγμα #7

Κατασκευάστε πρόγραμμα που θα βρίσκει τη γωνία που σχηματίζει με τον άξονα x ένα διάνυσμα με συντεταγμένες (x,y). Η γωνία να υπολογίζεται σε μοίρες και εντός του διαστήματος [0,360).

32

Παράδειγμα #7

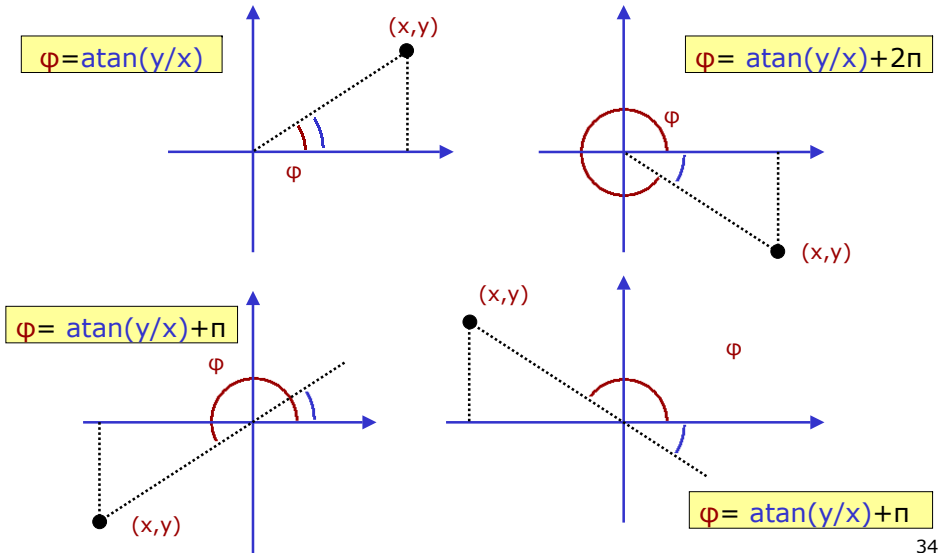


Η γωνία υπολογίζεται ως $\text{atan}(y/x)$. Όμως:

- Αν $x=0$ δεν μπορεί να γίνει η διαίρεση.
- Σημεία συμμετρικά ως προς την αρχή των αξόνων έχουν την ίδια εφαπτομένη.
- Η συνάρτηση atan επιστρέφει αποτέλεσμα στην περιοχή $-\pi/2$ έως $\pi/2$
- Το αποτέλεσμα επιστρέφεται σε ακτίνια και όχι σε μοίρες.

33

Παράδειγμα #7



34

Παράδειγμα #7

```
#include <iostream>
#include <cmath>
using namespace std;

int main ( )
{
    double x, y;           // Συντεταγμένες σημείου
    double pi;            // Ο αριθμός π
    double phi;           // Η γωνία

    cout << "Δώστε το σημείο ";
    cin >> x >> y;

    pi = acos(-1.0);
```

Συνεχίζεται...

35

Παράδειγμα #7

```
...Συνέχεια
if ( x==0 )
    if ( y > 0 )
        phi = 90;
    else if ( y < 0 )
        phi = 270;
    else
        phi = 0;
else
    if ( x > 0 )
        if ( y >= 0 )
            phi = atan(y/x)*180/pi;
        else
            phi = atan(y/x)*180/pi+360;
    else
        phi = atan(y/x)*180/pi+180;
cout << "Η γωνία είναι " << phi << " μοίρες\n";
}
```

36

ΥΠΟΛΟΓΙΣΤΕΣ ΙΙ

Εντολή switch

Τι κάνει η εντολή switch

Η εντολή switch ελέγχει αν η τιμή μιας ακέραιας μεταβλητής ισούται με κάποια σταθερά (από ένα πλήθος σταθερών) και ανάλογα εκτελεί συγκεκριμένες εντολές.

Συντακτικό της εντολής switch

```
switch ( ακέραια παράσταση ) {  
    case σταθερά1:  
        εντολή ;  
        εντολή ;  
        ⋮  
        break ;  
    case σταθερά2:  
        εντολή ;  
        εντολή ;  
        ⋮  
        break ;  
    default:  
        εντολή ;  
        εντολή ;  
        ⋮  
}
```

} Μπορεί να παραληφθεί

Πως λειτουργεί η εντολή switch

- Η εντολή switch αποτελείται από μια σειρά από εντολές και σημεία εισόδου (όπου υπάρχει case).
- Ελέγχεται η ακέραια παράσταση με τις αντίστοιχες σταθερές στις case, και προσδιορίζεται το σημείο εισόδου.
- Κατόπιν εκτελούνται όλες οι εντολές μέχρι το τελικό άγκιστρο, ή μέχρι να βρεθεί break.
- Εάν δεν ταιριάζει καμία case, τότε το σημείο εισόδου είναι το default.
- Το break μας βγάζει από την εντολή switch.

Παράδειγμα #8

Κατασκευάστε πρόγραμμα που θα υπολογίζει το
όγκο ενός από τα παρακάτω:
Ορθογώνιο παραλληλεπίπεδο
Σφαίρα
Κύλινδρος

Υπενθύμιση:

$$V_{\text{παρ}} = abc \quad V_{\text{σφ}} = \frac{4}{3}\pi r^3 \quad V_{\text{κυλ}} = \pi r^2 h$$

41

Παράδειγμα #8

```
#include <iostream>
#include <cmath>
using namespace std;

int main ( )
{
    int n; // Επιλογή σχήματος
    double a, b, c; // Πλευρές ορθογωνίου
    double pi; // Ο αριθμός π
    double r, h; // Για τον κύλινδρο
    double v; // Ο όγκος

    cout << "Επιλέξτε αντικείμενο: \n";
    cout << "1. Ορθογώνιο παραλληλόγραμμο \n";
    cout << "2. Σφαίρα \n";
    cout << "3. Κύλινδρος \n";
```

Συνεχίζεται...

42

Παράδειγμα #8

```
cin >> n;
switch (n) {
    case 1:
        cout << "Μήκη πλευρών ? \n";
        cin >> a >> b >> c;
        v = a*b*c;
        cout << "Ο όγκος είναι: " << v << endl;
        break;
    case 2:
        cout << "Ακτίνα σφαίρας? \n";
        cin >> r;
        pi = acos(-1.0);
        v = 4*pi*r*r*r/3;
        cout << "Ο όγκος είναι: " << v << endl;
        break;
```

Συνεχίζεται...

43

Παράδειγμα #8

```
case 3:
    cout << "Ακτίνα και ύψος ? \n";
    cin >> r >> h;
    pi = acos(-1.0);
    v = pi*r*r*h;
    cout << "Ο όγκος είναι: " << v << endl;
    break;
default:
    cout << "Λανθασμένη επιλογή. \n";
}
}
```

44