ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΥΠΟΛΟΓΙΣΤΩΝ & ΥΠΟΛΟΓΙΣΤΙΚΗ ΦΥΣΙΚΗ

Μέρος 4ο

ΝΙΚΟΛΑΟΣ ΣΤΕΡΓΙΟΥΛΑΣ

ΤΜΗΜΑ ΦΥΣΙΚΗΣ

ΑΡΙΣΤΟΤΕΛΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ

ΟΙ ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΗΣ

Με τους τελεστές σύγκρισης, συγκρίνουμε τις τιμές δύο μεταβλητών ή δύο εκφράσεων.

(a > b) (μεγαλύτερο)
(a < b) (μικρότερο)
(a >= b) (μεγαλύτερο ή ίσο)
(a <= b) (μικρότερο ή ίσο)
(a != b) (όχι ίσο)
(a == b) (ίσο)

ΠΡΟΣΟΧΗ: συγκρίνουμε την ισότητα με το διπλό == και όχι με το απλό = .

ΑΛΗΘΗΣ/ΨΕΥΔΗΣ ΕΚΦΡΑΣΗ

Όταν μια έκφραση στην οποία χρησιμοποιούμε τελεστή σύγκρισης είναι ΑΛΗΘΗΣ, τότε η τιμή της είναι 1.

π.χ. αν `i = 9;` τότε η `(i > 5)` έχει τιμή 1.

Όταν μια έκφραση στην οποία χρησιμοποιούμε τελεστή σύγκρισης είναι ΨΕΥΔΗΣ, τότε η τιμή της είναι 0.

π.χ. αν `i = 9;` τότε η `(i > 15)` έχει τιμή 0.

Επίσης, κάθε έκφραση με τιμή διαφορετική από 0 χαρακτηρίζεται αληθής, ενώ κάθε έκφραση με τιμή 0 χαρακτηρίζεται ψευδής.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int a = 3;

    printf("%d\n", (a>3) );
    printf("%d\n", (a<3) );
    printf("%d\n", (a>=3) );
    printf("%d\n", (a<=3) );
    printf("%d\n", (a==3) );
    printf("%d\n", (a!=3) );

    return 0;
}
```

```
0
0
1
1
1
0
```

ΣΥΝΔΥΑΣΤΙΚΟΙ ΤΕΛΕΣΤΕΣ

Η έκφραση

```
a  +=  b;
```

είναι μια συντομογραφία της έκφρασης

```
a  =  a + b;
```

Ο ίδιος κανόνας ισχύει και για τους τελεστές -, *, /, % .

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int a = 4, b = 2;

    a += 6;
    a *= b+3;
    a -= b+8;
    a /= b;
    a %= b+1;

    printf("Το αποτέλεσμα είναι = %d\n", a);

    return 0;
}
```

```c
a = a + 6;
a = a * (b+3);
a = a - (b+8);
a = a / b;
a = a % (b+1);
```

```
Το αποτέλεσμα είναι = 2
```

ΛΟΓΙΚΟΙ ΤΕΛΕΣΤΕΣ

Οι συνήθεις λογικοί τελεστές γράφονται ως:

AND : `&&`
OR : `||`

Παραδείγματα:

Η έκφραση `(10 == 10) && (5 > 3)` είναι αληθής, διότι και οι δύο επιμέρους εκφράσεις είναι αληθείς.

Η έκφραση `(10 == 10) || (5 < 3)` είναι αληθής, διότι έστω μία επιμέρους έκφραση είναι αληθής.

Ο ΤΕΛΕΣΤΗΣ sizeof

Υπολογίζει πόσα bytes δεσμεύονται στη μνήμη για κάθε μεταβλητή. Π.χ.

```c
#include <stdio.h>

int main(void)
{
    printf("char = %d bytes\n", sizeof(char));
    printf("int = %d bytes\n", sizeof(int));
    printf("float = %d bytes\n", sizeof(float));
    printf("double = %d bytes\n", sizeof(double));

    return 0;
}
```

```
char = 1 bytes
int = 4 bytes
float = 4 bytes
double = 8 bytes
```

Η ΕΝΤΟΛΗ if

Η if είναι από τις βασικότερες δομές ελέγχου της ροής ενός προγράμματος. Στην ποιο απλή μορφή, η σύνταξη είναι:

```
if ( συνθήκη )
{
  ...
  ...  /* ομάδα εντολών */
  ...
}
```

Αν `συνθήκη = ΑΛΗΘΗΣ`, τότε εκτελείται η ομάδα εντολών μέσα στα άγκιστρα.

Αν `συνθήκη = ΨΕΥΔΗΣ`, τότε δεν εκτελείται η ομάδα εντολών μέσα στα άγκιστρα.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int x = 3;


    if( x == 0 )
    {
        printf("Η μεταβλητή x είναι 0.\n");
    }


    if( x != 0 )
    {
        printf("Η μεταβλητή x δεν είναι 0.\n");
    }

    return 0;
}
```

```
Η μεταβλητή x δεν είναι 0.
```

Η ΕΝΤΟΛΗ if

ΣΗΜΕΙΩΣΗ: Αν η ομάδα εντολών μέσα στα άγκιστρα είναι μόνο μία εντολή, τότε τα άγκιστρα είναι προαιρετικά, δηλ. αρκεί:

```
if( x == 0 )
    printf("Η μεταβλητή x είναι 0.\n");

if( x != 0 )
    printf("Η μεταβλητή x δεν είναι 0.\n");
```

ΠΡΟΣΟΧΗ: Στο τέλος της συνθήκης του if δε βάζουμε ; !

(συντακτικό λάθος)

```
if( x == 0 );
    printf("Η μεταβλητή x είναι 0.\n");

if( x != 0 );
    printf("Η μεταβλητή x δεν είναι 0.\n");
```

Η ΕΝΤΟΛΗ if

ΠΡΟΣΟΧΗ: Στη συνθήκη της εντολής if, δεν πρέπει να συγχέουμε τον τελεστή ισότητας == με τον τελεστή εκχώρησης = .

```
#include <stdio.h>

int main(void)
{
    int x = 3;

    if ( x == 3)
       printf("Η τιμή του x είναι %d\n", x);

    if ( x = 2)
       printf("Η τιμή του x είναι %d\n", x);

    return 0;
}
```

(άλλαξε η τιμή του x)

```
Η τιμή του x είναι 3
Η τιμή του x είναι 2
```

Η ΕΝΤΟΛΗ if - else

Χρησιμοποιούμε το else όταν θέλουμε να εκτελεστεί μια συγκεκριμένη ομάδα εντολών στην περίπτωση που η συνθήκη ελέγχου στην if είναι ψευδής.

```
if ( συνθήκη )
{
  …
  …   /* ομάδα εντολών */
  …
}
else {
  …
  …   /* ομάδα εντολών */
  …
 }
```

(εκτελείται αν συνθήκη = ΑΛΗΘΗΣ)

(εκτελείται αν συνθήκη = ΨΕΥΔΗΣ)

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int x;

    printf("Δώσε έναν εκέραιο αριθμό:\n");
    scanf("%d",&x);

    if (x > 0)
        printf("Ο ακέραιος αριθμός είναι θετικός.\n");
    else
        printf("Ο ακέραιος αριθμός είναι αρνητικός ή μηδέν.\n");

    return 0;
}
```

```
Δώσε έναν εκέραιο αριθμό:
-4
Ο ακέραιος αριθμός είναι αρνητικός ή μηδέν.
```

ΠΑΡΑΔΕΙΓΜΑ: ΕΥΡΕΣΗ ΜΕΓΙΣΤΗΣ ΤΙΜΗΣ

```c
#include <stdio.h>

int main(void)
{
    double price1, price2, price3, price4, max;

    printf("Δώσε τέσσερις τιμές: ");

    scanf("%lf%lf%lf%lf", &price1, &price2, &price3, &price4);

    if(price1 > price2)

        max = price1;

    else

        max = price2;

    if(price3 > max)

        max = price3;

    if(price4 > max)

        max = price4;

    printf("Η μέγιστη τιμή είναι %f\n", max);

    return 0;

}
```

ΣΥΝΕΧΟΜΕΝΕΣ ΕΝΤΟΛΕΣ if - else

Μια εντολή if - else μπορεί να συνεχίζεται με νέα εντολή if - else, π.χ.

```
if ( συνθήκη )
{
   …
}
else if ( συνθήκη )
{
   …
}
else if ( συνθήκη )
{
   …
}
else
{
   …
}
```

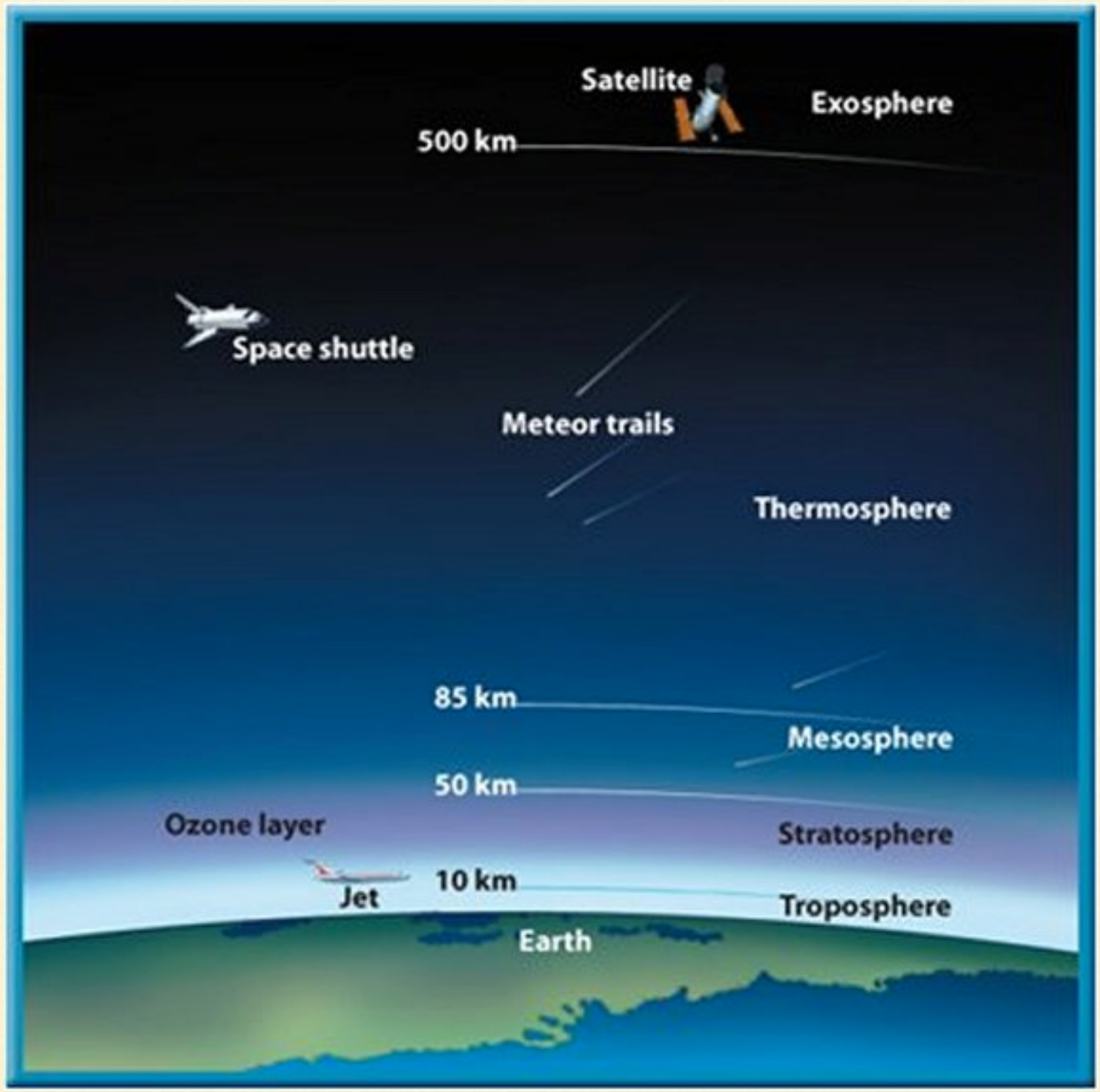
ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int x;

    printf("Δώσε έναν εκέραιο αριθμό:\n");
    scanf("%d",&x);

    if (x > 0)
        printf("Ο ακέραιος αριθμός είναι θετικός.\n");
    else if (x < 0)
        printf("Ο ακέραιος αριθμός είναι αρνητικός.\n");
    else
        printf("Ο ακέραιος αριθμός είναι το μηδέν.\n");

    return 0;
}
```

```
Δώσε έναν εκέραιο αριθμό:
0
Ο ακέραιος αριθμός είναι το μηδέν.
```

ΠΑΡΑΔΕΙΓΜΑ

Να γραφεί ένα πρόγραμμα που να υποδεικνύει την περιοχή της ατμόσφαιρας, με βάση το ύψος από το έδαφος.



ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    float height;

    printf("Δώσε το ύψος σε km: ");

    scanf("%f", &height);

    if(height == 0.0)
        printf("Έδαφος\n");

    else if(height > 0.0 && height < 10.0)

        printf("Τροπόσφαιρα\n");

    else if(height >= 10.0 && height < 50.0)

        printf("Στρατόσφαιρα\n");

    else if(height >= 50.0 && height < 85.0)

        printf("Μεσόσφαιρα\n");

    else if(height >= 85.0 && height < 500.0)

        printf("θερμόσφαιρα\n");

    else if(height >= 500.0)

        printf("Εξώσφαιρα\n");

    else
        printf("Το ύψος πρέπει να είναι θετικός αριθμός\n");

    return 0;

}
```

ΕΝΘΕΤΕΣ ΕΝΤΟΛΕΣ if - else

Γενικότερα, μια εντολή if - else μπορεί να περιέχει μέσα της μια ένθετη εντολή if - else, π.χ.

```
if ( συνθήκη )
{
    if ( συνθήκη )
    {
      …
    }
    else {
            …
            …
          }
}
else {
        …
     }
```

(ένθετη εντολή if - else)

ΕΝΘΕΤΕΣ ΕΝΤΟΛΕΣ if - else

Γενικότερα, μια εντολή if - else μπορεί να περιέχει μέσα της μια ένθετη εντολή if - else, π.χ.

```
if ( συνθήκη )
{
   …
}
else {
if ( συνθήκη )
{
  …
  …
}
else {
        …
        …
}
}
```

(ένθετη εντολή if - else)

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int a = 10, b = 20, c = 30;

    if(a > 5)
    {
        if(b == 20)
            printf("Ναι, b=20.\n");

        if(c == 40)
            printf("Ναι, c=40.\n");
        else
            printf("Όχι, c!=40.\n");
    }
    else
        printf("Όχι, a<=5.\n");

    return 0;
}
```

```
Ναι, b=20.
Όχι, c!=40.
```

Ο ΤΕΛΕΣΤΗΣ ?

Μια συντομογραφία της εντολής if-else είναι ο τελεστής ? Αντί να γράψουμε

```
if ( συνθήκη )
    εντολή_1;
else
    εντολή_2;
```

μπορούμε να γράψουμε πιο συντομα:

```
( συνθήκη ) ? εντολή_1 : εντολή_2;
```

Αν **συνθήκη = ΑΛΗΘΗΣ**, τότε εκτελείται η **εντολή_1**, αλλιώς, εκτελείται η **εντολή_2**.

ΠΑΡΑΔΕΙΓΜΑ

```c
#include <stdio.h>

int main(void)
{
    int a = 10, b = 20, c = 30;

    if(a > 5)
    {
        if(b == 20)
            printf("Ναι, b=20.\n");

        (c == 40) ? printf("Ναι, c=40.\n") : printf("Όχι, c!=40.\n");
    }
    else
        printf("Όχι, a<=5.\n");

    return 0;
}
```

```
Ναι, b=20.
Όχι, c!=40.
```

Ο ΤΕΛΕΣΤΗΣ ?

Το αποτέλεσμα μιας εφαρμογής του τελεστή ? μπορεί να εκχωρηθεί άμεσα ως τιμή σε μια μεταβλητή. Π.χ. αντί να γράψουμε

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

μπορούμε να γράψουμε πιο σύντομα:

```
max = ( a > b ) ? a : b;
```

ΣΗΜΕΙΩΣΗ: Μπορούμε να έχουμε ένθετες εφαρμογές του τελεστή ? αλλά αυτό δε συνίσταται, καθώς το πρόγραμμα γίνεται πολύ δυσανάγνωστο.

Η ΕΝΤΟΛΗ switch

Είναι εναλλακτική κάποιων συνεχιζόμενων εντολών if - else, π.χ.

```
if ( έκφραση == τιμή_1 )
{
  εντολή_1;
}
else if ( έκφραση == τιμή_2 )
{
  εντολή_2;
}
else if ( έκφραση == τιμή_3 )
{
  εντολή_3;
}
else
{
  εντολή_4;
}
```

```
switch ( έκφραση )
{
   case τιμή_1:
      εντολή_1;
   break;

   case τιμή_2:
      εντολή_2;
   break;

   case τιμή_3:
      εντολή_3;
   break;

   default:
    εντολή_4;
   break;
}
```

Η ΕΝΤΟΛΗ switch

ΠΡΟΣΟΧΗ: Η εντολή switch δέχεται μόνο ακέραια έκφραση ως όρισμα, π.χ.

```
if ( a == 1 )
{
  εντολή_1;
}
else if ( a == 2 )
{
  εντολή_2;
}
else if ( a == 3 )
{
  εντολή_3;
}
else
{
  εντολή_4;
}
```

```
switch ( a )
{
  case 1:
    εντολή_1;
  break;

  case 2:
    εντολή_2;
  break;

  case 3:
    εντολή_3;
  break;

  default:
   εντολή_4;
  break;
}
```

Η ΕΝΤΟΛΗ switch

Ανάμεσα στις εντολές case και break δε χρειάζονται άγκιστρα.

Η switch ελέγχει διαδοχικά τις ακέραιες τιμές. Μόλις βρεθεί η **`έκφραση`** να είναι ίση με μία από τις ακέραιες τιμές, εκτελούνται οι εντολές αυτής της περίπτωσης και με το break το πρόγραμμα εξέρχεται από την εντολή switch.

Εάν η **`έκφραση`** δεν ισούται με καμία από τις ακέραιες τιμές, τότε εκτελούνται οι εντολές της περίπτωσης default και το πρόγραμμα εξέρχεται από την εντολή switch. Η περίπτωση default δεν είναι υποχρεωτική.

ΠΑΡΑΔΕΙΓΜΑ: ΑΠΛΟ CALCULATOR

```c
#include <stdio.h>

int main()

{
    char operation;
    double a, b;

    printf("Δώσε την πράξη μεταξύ δύο αριθμών: ");

    scanf("%lf%c%lf", &a, &operation, &b);

    switch(operation)
    {
        case '+':
            printf("Άθροισμα = %f\n", a+b);
        break;

        case '-':
            printf("Διαφορά = %f\n", a-b);
        break;

        case '*':
            printf("Γινόμενο = %f\n", a*b);
        break;

        case '/':
            if(b != 0)
                printf("Πηλίκο = %f\n", a/b);
            else
                printf("Ο διαιρέτης δε μπορεί να είναι μηδέν.\n");
        break;

        default:
            printf("Μη αποδεκτή πράξη\n");
        break;
    }

    return 0;
}
```

(δε θα χρειαστεί κενά, π.χ. δίνουμε 3.0+4.1 και όχι 3.0 + 4.1 , διότι η ενδιάμεση μεταβλητή είναι char.)

ΠΑΡΑΔΕΙΓΜΑ: ΥΠΟΛΟΓΙΣΜΟΣ ΕΜΒΑΔΟΥ

```c
#include <stdio.h>

#define PI 3.14159265358979323846

int main(void)
{
    int selection;
    double length, radius;

    printf("Επίλεξε 0 για τετράγωνο, 1 για κύκλο: ");
    scanf("%d", &selection);

    switch(selection)
    {
        case 0:
            printf("Δώσε το μήκος της πλευράς: ");
            scanf("%lf", &length);
            if(length <= 0.0)
            {
                printf("Μη-αποδεκτό μήκος.\n");
                return 0;
            }
            printf("Το εμβδαδόν του τετραγώνου είναι %f\n", length*length);
        break;

        case 1:
            printf("Δώσε την ακτίνα ");
            scanf("%lf", &radius);
            if(radius <= 0.0)
            {
                printf("Μη-αποδεκτή ακτίνα.\n");
                return 0;
            }
            printf("Το εμβαδό του κύκλου είναι %f\n", PI*radius*radius);
        break;

        default:
            printf("Μη-αποδεκτή επιλογή.\n");
        break;
    }
    return 0;
}
```

ΣΥΝΕΝΩΣΗ cases

Αν οι εντολές που αντιστοιχούν σε δύο ή περισσότερες περιπτώσεις case είναι κοινές, μπορεί να γίνει συνένωση, π.χ.

```
switch ( a )
{
   case 1:
   case 2:
   case 3:
      εντολή_1;
   break;

   case 4:
   case 5:
      εντολή_2;
   break;

   default;
    εντολή_3;
   break;
}
```