ΕΠΛ 033: ΕΙΣΑΓΩΓΗ ΣΤΟΝ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ ΓΙΑ ΜΗΧΑΝΙΚΟΥΣ

Μάριος Belk, Τμήμα Πληροφορικής, Πανεπιστήμιο Κύπρου

Email: belk@cs.ucy.ac.cy

ΔΟΜΕΣ ΕΠΙΛΟΓΗΣ

Θέματα Διάλεξης – Δομές Επιλογής

- *Σχεσιακοί και Λογικοί Τελεστές Λογικές Παραστάσεις*
- *Εντολές if, if-else, φωλιασμένα if-else, ? :*
- *Εντολή switch*
- *Έλεγχος ορθότητας δεδομένων και συναρτήσεων*

Δομές Εκτέλεσης

- Διαδοχική εκτέλεση
  - Στηρίζεται στην απλή παράθεση εκφράσεων/εντολών, η μια μετά την άλλη
- Εκτέλεση με επιλογή
  - Η ροή του προγράμματος “διακόπτεται” για να παρθεί μια απόφαση, να γίνει κάποια επιλογή
  - Το αποτέλεσμα της απόφασης καθορίζει την “κατεύθυνση” της ροής του προγράμματος
- Εκτέλεση με επανάληψη
  - Μια ομάδα εκφράσεων / εντολών εκτελείται περισσότερο από μια φορά

Δομές Επιλογής (Selection)

- Έκφραση Συνθήκης
  - Σχεσιακοί/Συγκριτικοί Τελεστές
  - Λογικοί Τελεστές
- Εντολές if, if-else, ? :
  - Φωλιασμένα if-else
- Εντολή switch

Παράδειγμα Επιλογής

- Γράψτε κώδικα που παίρνει παραμέτρους δυο ακέραιους αριθμούς και τυπώνει τον μικρότερο
- Λύση (ψευδοκώδικας):

διάβασε τους ακέραιους α, β

αν (α < β) **τότε**

    τύπωσε α

αλλιώς

    τύπωσε β

Συνθήκη στις δομές επιλογών

αν (συνθήκη) **τότε**

εντολές Α

αλλιώς

εντολές Β

συνθήκη : λογική παράσταση που αποτιμάται σε **true (1)** ή **false (0)**

Λογικές παραστάσεις

- Μια **λογική παράσταση (ΛΠ)** είναι ανάλογη μαθηματικής παράστασης, με την διαφορά ότι **το αποτέλεσμα μπορεί να είναι μόνο αλήθεια (1) ή λάθος (0)**
- Λογικές παραστάσεις συνθέτονται χρησιμοποιώντας σχεσιακούς τελεστές
- Δυο λογικές παραστάσεις μπορούν να συνδυαστούν με ένα λογικό τελεστή

Σχεσιακοί Τελεστές (Relational Operators)

- Δυαδικοί Τελεστές
  - **<** μικρότερο από
  - **>** μεγαλύτερο από
  - **<=** μικρότερο ή ίσο με
  - **>=** μεγαλύτερο ή ίσο με
  - **==** ίσο με
  - **!=** διάφορο του
- Αποτιμούνται σε 0 (ψευδές) ή 1 (αληθές)
- Τύποι τελεστέων int, char, float, double

Παραδείγματα

- (x < y)
- t = (x < y); Αν (t) τότε printf(“true\n”);
- k = (i >= 8);
- a = (b != b);
- (f == 2.3456)
- (‘a’ >= ‘b’)
  - ‘a’<‘b’<..<‘z’
  - ‘A’<‘B’<..<‘Z’
  - ‘0’<‘1’<..<‘9’

Λογικοί Τελεστές (Logical Operators)

- Συνδυάζουν δύο λογικές παραστάσεις σε μια σύνθετη λογική παράσταση
  - **&&** σύζευξη, δυαδικός τελεστής (**AND**)
  - **||** διάζευξη, δυαδικός τελεστής (**OR**)
  - **!** άρνηση, μοναδιαίος τελεστής (**NOT**)
- Αποτιμούνται σε 0 ή 1
  - **0** (δεν ισχύει, ψευδές ή **false**)
  - **1** (ισχύει, αληθές ή **true**)

Εκφράσεις με Λογικούς και Σχεσιακούς Τελεστές

```
t = (A>=0 && A<=100);

y = (s<10 || s>100);

!(s<10 || s>100)

(i<10  && j==1)
```

Λογικές εκφράσεις σε μεταβλητές

- t = (A>=0 && A<=100);

  t1 = (A>= 0);

  t2 = (A <= 100);

  t = (t1 && t2);

Πίνακας Αληθείας για && (AND)

Τελεστέος A	Τελεστέος B	A && B
0	0	0
0	1	0
1	0	0
1	1	1

Όταν χρειάζεται να ελέγξουμε πως όλες οι υποεκφράσεις ισχύουν, π.χ. (a && b)

Πίνακας Αληθείας για || (OR)

Τελεστέος A	Τελεστέος B	A || B
0	0	0
0	1	1
1	0	1
1	1	1

Όταν χρειάζεται να ελέγξουμε πως τουλάχιστον μια υποέκφραση ισχύει, π.χ. (a || b)

Πίνακας Αληθείας για ! (NOT)

Τελεστέος Α	!Α
0	1
1	0

Όταν χρειάζεται να ελέγξουμε πως ισχύει το αντίθετο μιας έκφρασης π.χ. !(a && b)

Συμπλήρωμα Συνθηκών

- **Το συμπλήρωμα μίας συνθήκης είναι η άρνηση της**, π.χ.

Συνθήκη	**Συμπλήρωμα**
1	0
0	1
item == SENT	!(item == SENT) ή item != SENT
item > SENT	item <= SENT
(n % 2) == 0	(n % 2) != 0

Κανόνες DeMorgan

- $!(E_1 \&\& E_2) \equiv !E_1 \ || \ !E_2$
- $!(E_1 \ || \ E_2) \equiv !E_1 \ \&\& \ !E_2$

```
!(age > 25  &&  (status == 'S' || status == 'D'))
≡ !(age > 25) ||  !(status == 'S' ||  status == 'D')
≡ age <= 25  || (status != 'S' &&  status != 'D')
```

Προτεραιότητες Τελεστών

υψηλότερη

- **( )**
- **! & (μοναδιαίοι τελεστές)**
- *** / %**
- **+ –**
- **< <= >= >**
- **== !=**
- **&&**
- **||**
- **=**

χαμηλότερη

Αποτίμηση Λογικών Εκφράσεων

- Η αποτίμηση αρχίζει από τα αριστερά και προχωρεί μέχρι το σημείο που χρειάζεται να προχωρήσει (lazy evaluation), π.χ.
  - 0 **&&** E → 0
    - η αποτίμηση της έκφρασης E δεν χρειάζεται
  - 1 **||** E 1
    - η αποτίμηση της έκφρασης E δεν χρειάζεται

Σειρά Συνθηκών σε μία Λογική Έκφραση

- Για λόγους αποδοτικότητας
- Αποφυγή λαθών
  - (y / **x**) > 2 && x != 0
  - Εάν x ισούται με 0 → run time error (σφάλμα κατα την εκτέλεση του προγράμματος)
  - Η ακόλουθη διατύπωση δεν έχει πρόβλημα, λόγω της σειράς των συνθηκών
    - x != 0 && (y / **x**) > 2

Σειρά Συνθηκών σε μία Λογική Έκφραση (συν.)

- Γενικά σε μία σύζευξη, E1 && E2, η συνθήκη E1 πρέπει να αποτελεί τον αριστερό τελεστέο, εάν στην περίπτωση που αποτιμάται σε 0, τυχόν αποτίμηση της E2 θα οδηγήσει σε πρόβλημα. Με αυτή την σειρά αποτρέπεται η αποτίμηση της E2.

- Ανάλογο για διάζευξη E1||E2

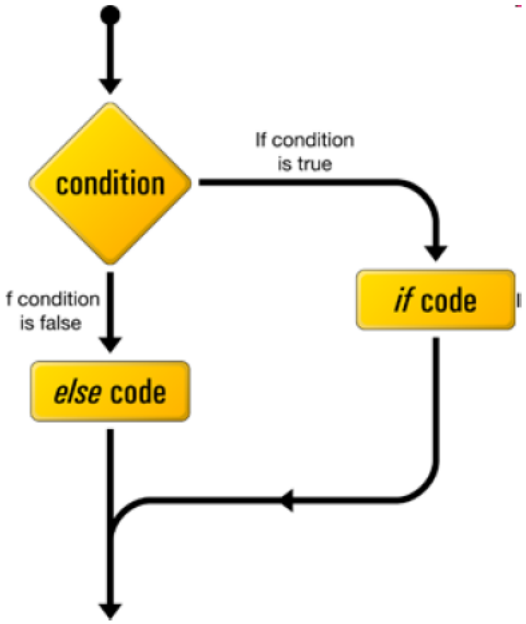
Μετατροπή Εκφράσεων σε C

- x και y μεγαλύτερα του z
  - (x>z && y>z)
- x είναι ίσο με το 2 ή με το 10
  - (x==2 || x==10)
- a είναι στο πεδίο από b μέχρι και c
  - (a>=b && a<=c)
- a είναι έξω από το πεδίο b μέχρι και c
  - !(a>=b && a<=c) ή (a<b || a>c)
- x είναι μικρότερο του 0 ή μεταξύ 10 και 1000
  - (x<0) || (x>=10 && x<=1000)
- z είναι αγγλικός χαρακτήρας
  - (z>='a' && z<='z') || (z>='A' && z<='Z')

Εντολή διακλάδωσης if-else

Συντάσσεται ως εξής:

```
if (condition)
{
        ... // block A
}
else
{
        ... // block B
}
```



Εάν η συνθήκη (condition) είναι αληθής, εκτελείται το block των εντολών που περικλείεται μεταξύ των πρώτων {} (block A).

Αλλιώς, εκτελείται το block των εντολών μετά το **else** (block B).

Εντολή if-else

Απλές Εντολές

```
if (έκφραση)
    εντολή;
else
    εντολή;
```

Σύνθετες Εντολές

```
if (έκφραση){
    εντολή;
    ……..
}
else {
    εντολή;
    ……..
}
```

- Σύνθετες Εντολές (ή blocks)
  - { Δήλωση; …; Εντολή;}
  - Άγκιστρα ομαδοποιούν εντολές σαν να είναι μια εντολή

Σημασία if-else

- Σημασία: Εάν το αποτέλεσμα της έκφρασης (συνθήκης) είναι αληθές, τότε εκτελούνται οι εντολές του if-block αλλιώς εκτελούνται οι εντολές του else-block

```
if (temp > 0){
  printf("above freezing.\n");
}
else
{
  printf("freezing!\n");
}
```

Παράδειγμα 1: if-else

- Γράψτε κώδικα που παίρνει παραμέτρους δυο ακέραιους αριθμούς και τυπώνει τον μικρότερο

Παράδειγμα 1: if-else – Λύση

```
int main(){
    int a, b;

    printf("dwse ta a kai to b\n");
    scanf("%d%d", &a, &b);

    if (a < b)
        printf("%d\n",  a);
    else
        printf("%d\n",  b);

return 0;
}
```

Εντολή if

- if (έκφραση)
  εντολή;

```
if (έκφραση) {
    εντολή;
    ...
}
```

Σημασία if

- Σημασία: Εάν το αποτέλεσμα της έκφρασης (συνθήκης) είναι αληθές, τότε εκτελούνται οι εξαρτώμενες εντολές αλλιώς η εκτέλεση συνεχίζει μετά το if block (αγνοούνται οι εξαρτώμενες εντολές)

```
if (count <= 0){
    count = count + 1;
    printf("increment count\n");
}
printf("I am here!\n");
```

Παράδειγμα if

```
int number, count;
…..
if (number <= 0)   {
     count = count + 1;
}

printf("%d\n", count);
….
```

number	count	print output
2	3	?

Παράδειγμα if-else

- Γράψετε τη συνάρτηση **min()** που παίρνει παραμέτρους δυο ακέραιους αριθμούς και επιστρέφει τον μικρότερο εξ αυτών.

Παράδειγμα 2: if-else – Λύση

```
int min(int a, int b){
    int minimum;
    if (a < b)
        minimum = a;
    else
        minimum = b;

    return minimum;
}
```

Ενναλακτικές Λύσεις

```
int min(int a, int b){
    int minimum;
    if (a < b)
        minimum = a;
    else
        minimum = b;
    return minimum;
}
```

```
int min(int a, int b)
{
    if (a < b)
        return a;
    else
        return b;
}
```

```
int min(int a, int b){
    if (a < b)
        return a;
    return b;
}
```

```
int min(int a, int b){
    int minimum;
    minimum = b;

    if (a < b)
        minimum = a;

    return minimum;
}
```

Παράδειγμα if-else

- Γράψτε τη συνάρτηση min_three() η οποία παίρνει σαν παράμετρους τρεις ακέραιους αριθμούς και επιστρέφει τον μικρότερο εξ αυτών

Παράδειγμα if-else - Λύση

```
int min_three(int a, int b, int c){
    int minimum;
    minimum = min(a,b);
    minimum = min(minimum,c);

    return minimum;
}
```

Με χρήση της συνάρτησης **min()** την οποία υλοποιήσαμε στο Παράδειγμα 2

Ενναλακτική Λύση

```
int min_three(int a, int b, int c){
        int minimum;
        if (a < b)
                minimum = a;
        else
                minimum = b;
        if (c < minimum)
                minimum = c;
        return minimum;
}
```

Χωρίς τη χρήση οποιασδήποτε συνάρτησης

Φώλιασμα (nesting)

```
int min_three(int a, int b, int c){
        int minimum;
        if (a < b)
          if (a < c)
             minimum = a;
          else
             minimum = c;
        else
           if (b < c)
              minimum = b;
           else
              minimum = c;

        return minimum;
}
```

Παράδειγμα Φωλιασμένου if-else

- Γράψτε κώδικα που ελέγχει ένα ακέραιο αριθμό και τυπώνει
  - 1 εάν είναι θετικός,
  - 0 εάν είναι 0 και
  - -1 εάν είναι αρνητικός

Κώδικας 1

```
int detect(int number){
        int code;
        if (number > 0)
                code = 1;
        else
                if (number < 0)
                        code = -1;
                else /* number einai 0 */
                        code = 0;

        return code;
}
```

Κώδικας 2

```
int detect(int number){
    int code;
    if (number > 0){
        code = 1;
    }
    else{           /* number <= 0 */
        if (number < 0){
            code = -1;
        }
        else{  /* number = 0 */
            code = 0;
        }
    }
    return code;
}
```

Κώδικας 3 (if-else-if)

```
int detect(int number){
    int code;
    if (number > 0)
        code = 1;
    else if (number < 0)
        code = -1;
    else /* number einai 0 */
        code = 0;
return code;
}
```

Φίλτρα/ Έλεγχος Δεδομένων

- Να χρησιμοποιείτε τις εντολές ελέγχου για να ελέγχετε την ορθότητα δεδομένων π.χ.

```
if (size<=0){
        printf("Error: size is not positive\n");
        exit(-1); /*termatizei programma*/
}
```

- ή το assert (assert.h)

```
assert(size>0); /*Error: size is not positive - termatizei*/
```

Έλεγχος Συναρτήσεων

- Να χρησιμοποιείτε τις εντολές ελέγχου/ assert για να ελέγχετε την ορθότητα της εκτέλεσης συναρτήσεων βιβλιοθήκης

  - **Η συνάρτηση scanf επιστρέφει τον αριθμό αντικειμένων εισόδου που συμφώνησαν με την προδιαγραφή**

```
fields_read = scanf("%d", &num);
if (fields_read!=1) { /* error */ }
```

Παραδείγματα

```
printf("Enter length and width in meters: ");
field_read = scanf("%f %f", &length, &width);
if (field_read !=2){
    printf("Error: input in incorrect format\n");
    exit(-1);
    }
if (length<0){
    printf("Error: Length is not possitive!\n");
    assert(0);
    }
assert(length<10000); /* length is greater than
9999*/
```

Εντολή ? :

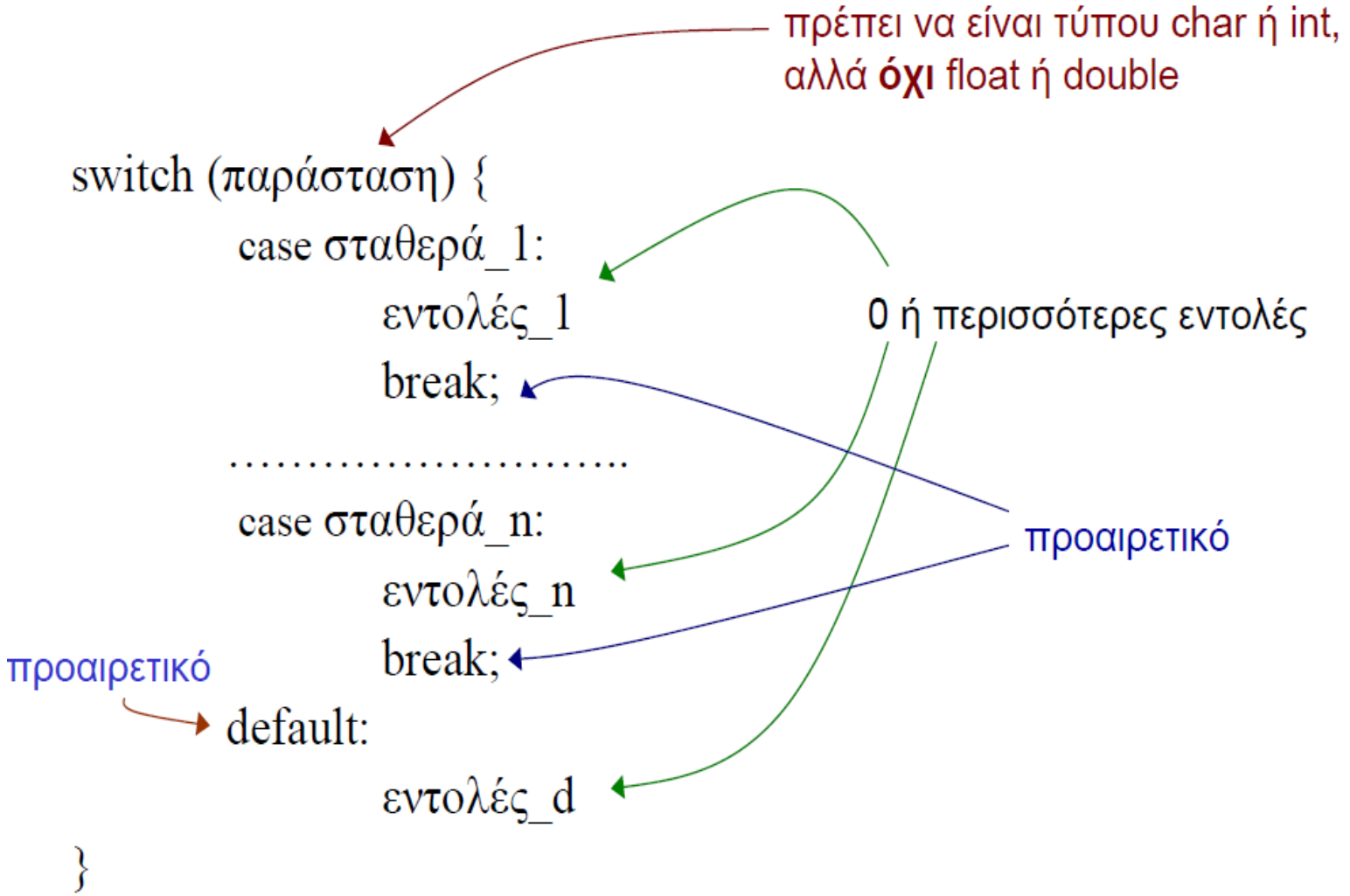
- Σύνταξη:

  (παράσταση) ? τιμήΑ : τιμήΒ;

- Σημασία: Εαν η τιμή της παράστασης είναι διάφορη του μηδενός τότε το αποτέλεσμα είναι
  - ***τιμήΑ*** αλλιώς
  - είναι ***τιμήΒ***

π.χ.: x = (y >100) ? y : 0;

Εντολή switch



Παράδειγμα εντολής switch

```
char letter_grade;
int counta, countb, count_other;
counta = countb = count_other = 0;
……
switch(letter_grade){
        case 'A':
                    counta=counta+1;
                    printf("excellent\n");
                    break;
         case 'B':
                     countb=countb+1;
                     printf("very_good\n");
                     break;
         case 'C':
         case 'D':
         case 'F':
                   count_other=count_other+1;
                   printf("other\n");
                   break;
         default:
                   printf("%c is the wrong grade!\n", letter_grade);
}
```

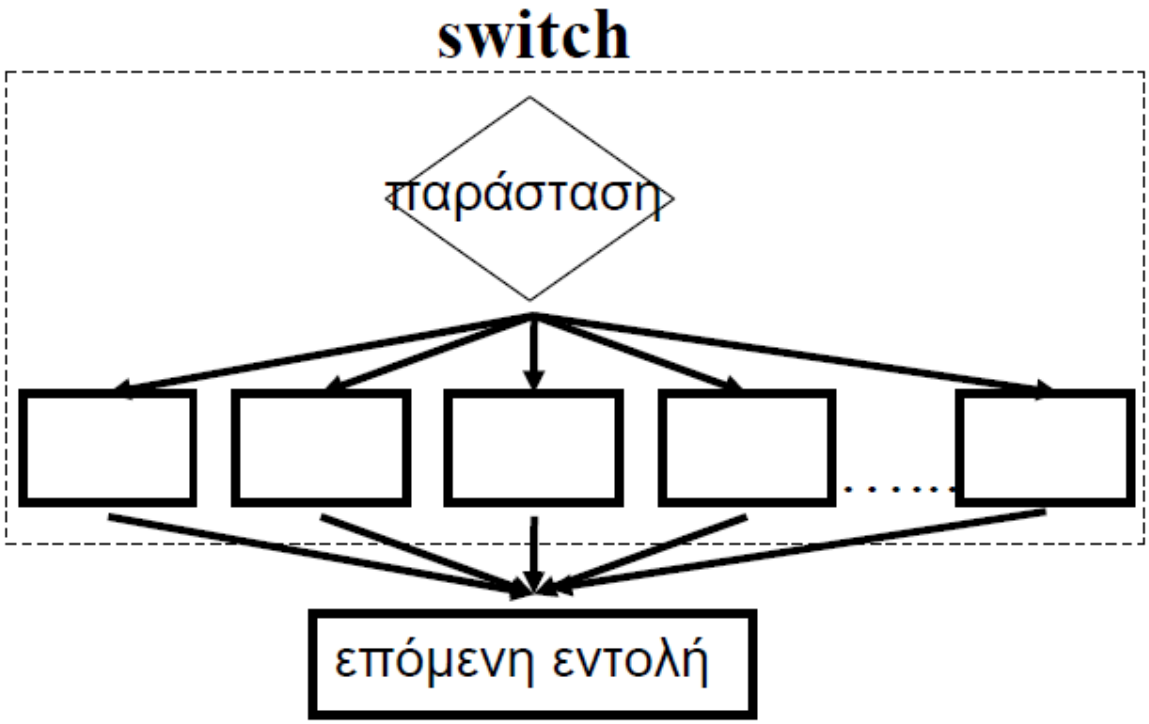
```
#include <stdio.h>
 int main () {

  /* local variable definition */
  char grade = 'B';
```

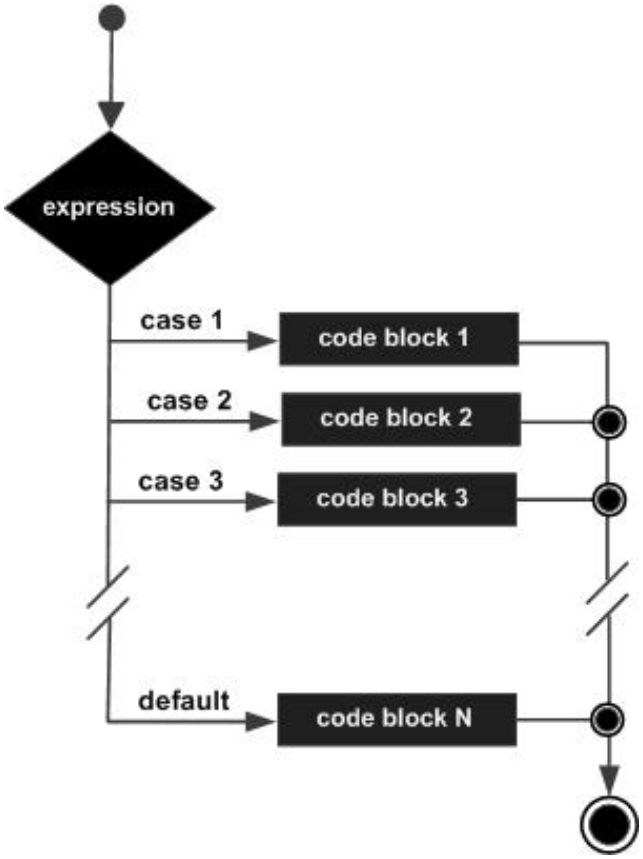


```
  switch(grade) {
     case 'A' :
     printf("Excellent!\n" );
        break;
     case 'B' :
     case 'C' :
     printf("Well done\n" );
        break;
     case 'D' :
     printf("You passed\n" );
        break;
   case 'F' :
     printf("Better try again\n" );
        break;
     default :
     printf("Invalid grade\n" );
  }
  printf("Your grade is  %c\n", grade );
  return 0;
}
```

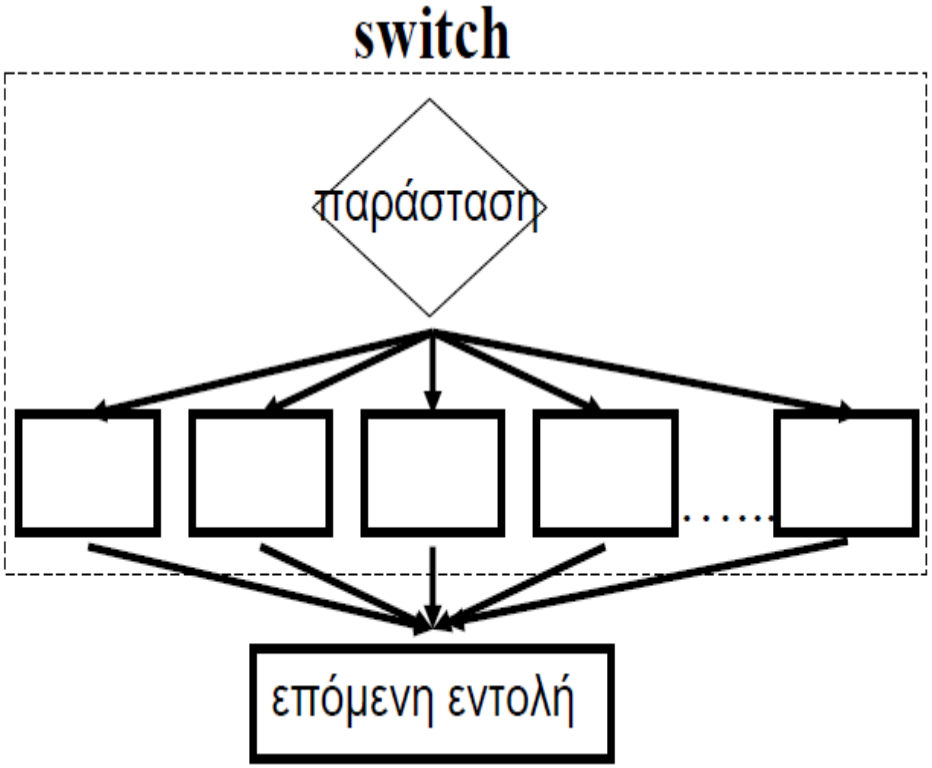
```
Well done
Your grade is B
```

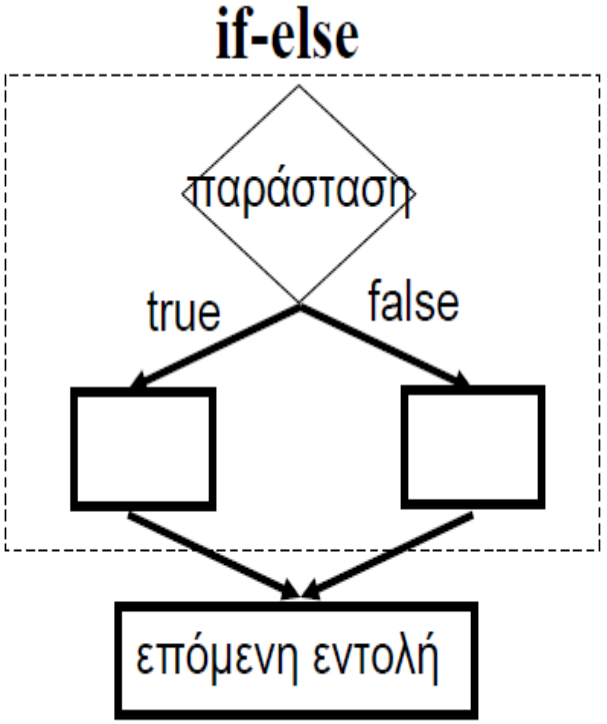
Ροή Ελέγχου με switch



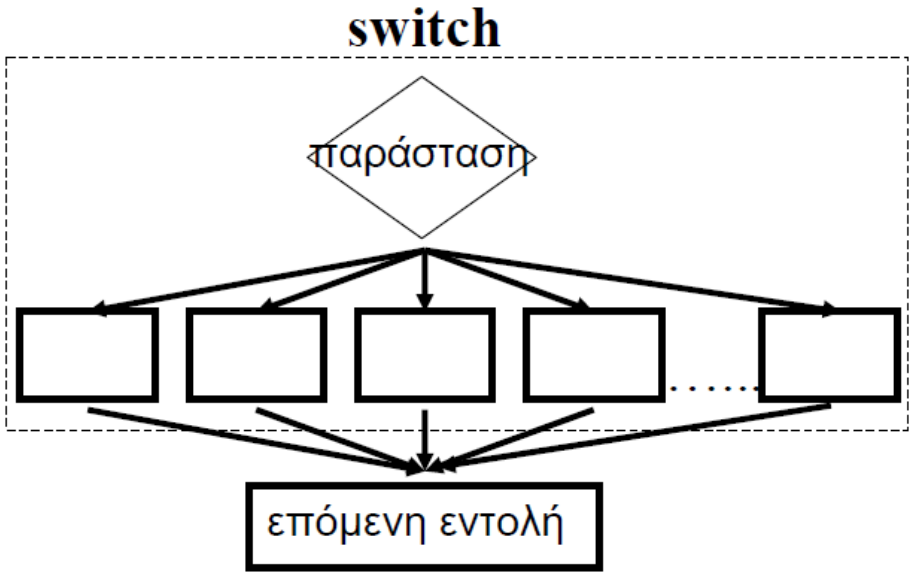


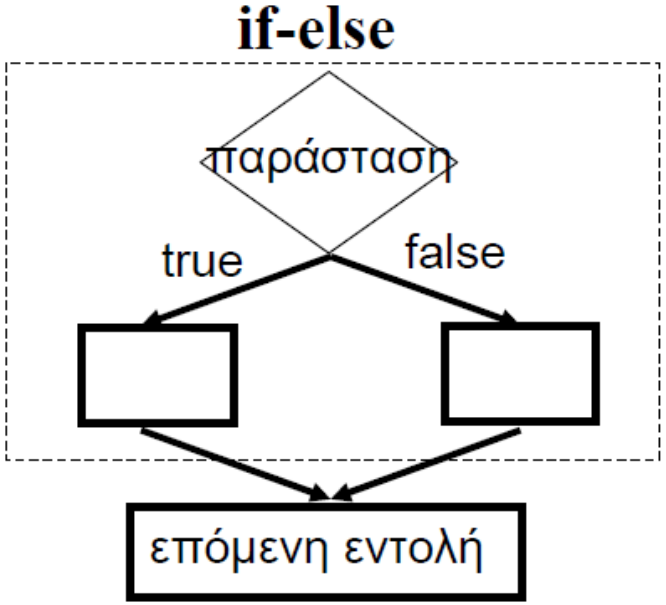
Switch vs if-else

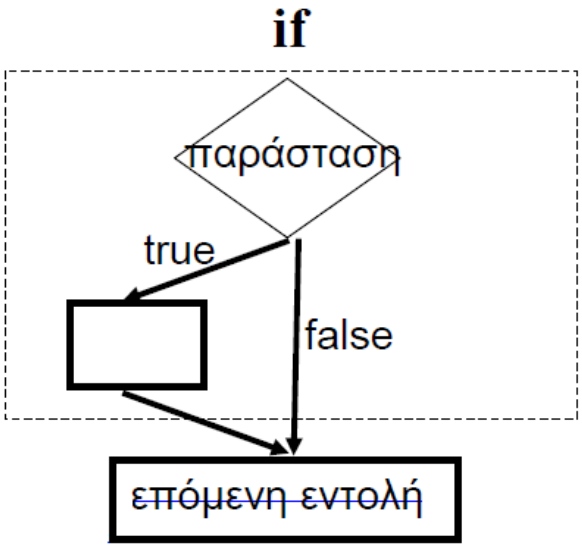




Switch, if-else, if







Παράδειγμα

- Γράψε κώδικα που διαβάζει ένα χαρακτήρα και τυπώνει 0 εάν ο χαρακτήρας είναι άσπρο διάστημα (whitespace) και 1 αλλιώς

- Τι είμαι άσπρο διάστημα;

Παράδειγμα – Λύση

```
#define WHITE_SPACE      0
#define NOT_WHITE_SPACE  1
    char c;
    int code;

    scanf("%c", &c);

    if (c== ' ' || c=='\n' || c=='\t')
          code = WHITE_SPACE;
    else
          code = NOT_WHITE_SPACE;
    printf("%d\n", code);
```

- **Input:** 123 3#4%^&* asdf
- **Output:** 11101111111001111

Άσκηση για εξάσκηση: Χρησιμοποιείστε switch αντί if-else

Παράδειγμα

- Γράψτε την διαδικασία display_grade_message που παίρνει τον βαθμό ενός φοιτητή (0 – 100), και τυπώνει τα μηνύματα:
  - πάνω απο 85 Excellent
  - 65-84 Very Good
  - 50-64 Good
  - κάτω από 50 Failure

Κώδικας

```
void display_grade_message(int weighted_aver)
{
    if (weighted_aver >= 85)
        printf("\nExcellent");
    else if (weighted_aver < 85 && weighted_aver >= 65)
        printf("\nVery Good");
    else if (weighted_aver < 65 && weighted_aver >= 50)
        printf("\nGood");
    else
        printf("\nFailure");
}
```

Άσκηση για εξάσκηση: Χρησιμοποιείστε switch αντί if-else

Κώδικας με switch


```
void display_grade_message(int weighted_aver){
     switch(weighted_aver)
     {
        case 85 ... 100:
            printf("\nExcellent");
            break;
        case 65 ... 84:
            printf("\nVery Good");
            break;
        case 50 ... 64:
            printf("\nGood");
            break;
        default:
            printf("\nFailure");
        }
}
```

Αυτό είναι ορθό;

```
void display_grade_message(int weighted_aver)
{
    if (weighted_aver >= 85)
          printf("\nExcellent");
    else if (weighted_aver >= 65)
          printf("\nVery Good");
    else if (weighted_aver >= 50)
          printf("\nGood");
    else
          printf("\nFailure");

}
```

Αυτό;

```
void display_grade_message(int weighted_aver){

    if (weighted_aver >= 85)
          printf("\nExcellent");
    if (weighted_aver >= 65)
          printf("\nVery Good");
    if (weighted_aver >= 50)
          printf("\nGood");
    if (weighted_aver < 50)
          printf("\nFailure");

}
```

Αυτό είναι ορθό;

```
void display_grade_message(int weighted_aver)
{
    if (weighted_aver < 50)
            printf("\nFailure");
    else if (weighted_aver >=50)
            printf("\nGood");
    else if (weighted_aver >= 65)
             printf("\nVery Good");
    else if (weighted_aver >= 85)
             printf("\nExcellent");
}
```

Αυτό είναι ορθό;

```
void display_grade_message(int weighted_aver)
{
    if (weighted_aver < 50)
            printf("\nFailure");
    else if (weighted_aver < 65)
            printf("\nGood");
    else if (weighted_aver < 85)
            printf("\nVery Good");
    else
            printf("\nExcellent");
}
```

Παράδειγμα

- Ένας χρόνος είναι δίσεκτος εάν διαιρείται ακριβώς με το 4, εκτός και εάν διαιρείται ακριβώς με το 100, στην οποία περίπτωση θα πρέπει να διαιρείται ακριβώς και με το 400

Σχεδιασμός Λύσης: Φάση 1

Ένας χρόνος είναι δίσεχτος **εάν διαιρείται ακριβώς με το 4**, εκτός και εάν διαιρείται ακριβώς με το 100 στην οποία περίπτωση θα πρέπει να διαιρείται ακριβώς και με το 400.

αν (x%4 == 0) **τότε**

είναι δίσεχτος

αλλιώς

δεν είναι

Σχεδιασμός Λύσης: Φάση 2

Ένας χρόνος είναι δίσεχτος εάν διαιρείται ακριβώς με το 4, **εκτός και εάν διαιρείται ακριβώς με το 100**, στην οποία περίπτωση θα πρέπει να διαιρείται ακριβώς και με το 400.

αν (x%4 == 0) **τότε**
    αν (x%100 == 0) **τότε**
        δεν είναι
    αλλιώς
        είναι δίσεχτος
αλλιώς
    δεν είναι

Σχεδιασμός Λύσης: Φάση 3

Ένας χρόνος είναι δίσεχτος εάν διαιρείται ακριβώς με το 4, εκτός και εάν διαιρείται ακριβώς με το 100 στην οποία περίπτωση θα **πρέπει να διαιρείται ακριβώς και με το 400.**

αν (x%4 == 0) **τότε**

αν (x%100 == 0) **τότε**

αν (x%400 == 0) **τότε**

είναι δίσεχτος

αλλιώς

δεν είναι

αλλιώς

είναι δίσεχτος

αλλιώς

δεν είναι

Κώδικας 1

```
void  is_leap (int year){
      int disektos;
      if (year % 4 == 0){
               if (year % 100  == 0) {
                        if (year % 400 == 0)
                                 disektos = 1;
                        else
                                 disektos = 0;
                        }
               else
                        disektos = 1;
      }
      else
               disektos = 0;
      printf("o xronos %s disektos\n ", disektos? "einai" : "den einai");
}
```

Κώδικας 2

```
void  is_leap (int year){
     int disektos;
     if (year%4==0) {
            if (year%100==0 && year%400==0)
                   disektos = 1;
            else if (year%100 != 0)
                   disektos = 1;
            else
                   disektos = 0 ; // Poia periptosi einai afti?
     }
     else
            disektos = 0;

     printf("o xronos %s disektos\n ", disektos? "einai" : "den
einai");
}
```

if-else και switch

- Οποιαδήποτε εντολή switch μπορεί να γραφεί με if-else (ισχύει και το αντίθετο)
- Επιλέξετε το πιο “φυσιολογικό”, “εύκολο” κτλ

Υλοποίηση με switch

```
switch (year % 4 == 0) {
    case 0:       disektos = 0;
          break;
    case 1:       switch (year % 100 == 0) {
          case 0:       disektos = 1;
                break;
          case 1:       switch (year % 400 == 0) {
                case 0:       disektos = 0;
                      break;
                case 1:       disektos = 1;
                      break;
                }
          }
    }
```

Κοινά Συντακτικά/Λογικά Λάθη

- = αντί ==
  - if (x=10) αντί if (x==10)
- συνθήκη χωρίς παρένθεση
  - if x>4 αντί if (x>4)
- Μετατροπή από μαθηματικά σε C, από 0 μέχρι 4:
  - (0 <= x <= 4) αντί (0 <= x && x<= 4)

Λάθος αντιστοιχία if-else

Λάθος:

```
if (x > 0)
        sum = sum + x;
        printf(”Greater than 0”);
else
        printf(”Less than 0”);
```

Ορθό:

```
if (x > 0){
        sum = sum + x;
        printf(”Greater than 0”);
}
else
        printf(”Less than 0”);
```

Άσκηση για εξάσκηση

Γράψτε πρόγραμμα για τον υπολογισμό του λογαριασμού ύδρευσης για οικιακή (Ο), επαγγελματική (Ε) και βιομηχανική χρήση (Β), με βάση τους κανόνες:

- Ο: πάγιο $5, σύν $0.5 ανά κυβικό μέτρο
- Ε: πάγιο $100, σύν $0.2 ανά κυβικό μέτρο μετά τα 1000 κυβικά
- Β: πάγιο $500, σύν $0.1 ανά κυβικό μέτρο μετά τα 5000 κυβικά

Σημείωση: Ο υπολογισμός θα γίνεται από την συνάρτηση **compute()**. Η συνάρτηση θα δέχεται σαν είσοδο τις πιο κάτω 2 παραμέτρους και θα τυπώνει το αποτέλεσμα στην οθόνη.

- Τον τύπο του λογαριασμού
- Τον αριθμό των κιλοβατώρων

Περίληψη

- Δομές Εκτέλεσης
- Λογικές Παραστάσεις
  - Σχεσιακοί και Λογικοί Τελεστές
  - Προτεραιότητα τελεστών
- Δομές Επιλογής
  - Εντολές if, if-else, φωλιασμένα if-else
  - Εντολή switch
- Έλεγχος Ορθότητας Δεδομένων και Συναρτήσεων
  - if και exit(-1)
  - if, scanf και fields_read
  - assert (χρήση βιβλιοθήκης assert.h)