SCIENCE AND TECHNIQUES DEPARTMENT 1st YEAR ST

# course 2: control structures

Dr. Samira LAGRINI

lagrini.samira83@gmail.com

Année universitaire:2022/2023

# Introduction

- Program instructions are generally executed in a sequential manner.
- However, in several cases, it is required either to choose between two or more execution choices or to repeat the execution of a set of instructions.
- Control structures make it possible to modify the behavior of a program following the fulfillment of one or more conditions,
- Among these structures we will see in this course: conditional structures and repetitive structures (loops).

# if….else   statement

- These structures are used when we have instructions or blocks of instructions which must only be executed if certain conditions are verified.

# if….else   statement

In this type of test one of the two blocks of instructions are executed depending on the satisfaction or not of a condition (logical expression which can be true or false).
The general form is as follows:

```
if ( condition )
        instruction-1
 else
        instruction-2
```

- If the condition is true then instruction-1 is executed, otherwise it is instruction -2

# if… statement

- If the program does not require performing an action if the condition is not verified, the else block can be deleted.
- The general form is as follows:

```
if ( condition )
        { instructions block }
```

if the condition is not verified, the program immediately proceeds to the next block of instructions delimited by the braces { }.

- Example : Maximum of two numbers

```
# include <stdio.h>
main()
{ float x,y;
printf("saisir deux valeurs réelles");
scanf ( %f %f , &x, &y);
  if (x<y)
      printf (« The maximum est : %f", y)
 else
      printf (« The maximum est: %f ", x);
}
```

**Example:** solving an equation   ax2+bx+c = 0  such as  a ≠ 0

```
#include <stdio.h>
#include <math.h>
 main()
 {   float a,b,c,x1,x2,delta;
    printf("Enter the coefficients a,b et c\n");
    scanf("%f%f%f",&a,&b,&c);
     delta = pow(b,2)-4*a*c;
       if (delta > 0 )
      { x1 = (-b-sqrt (delta)/(2*a));
        x2 = (-b+sqrt (delta)/(2*a));
       printf(" x1 %f \n x2 %f", x1,x2);
       }
       else
         if (delta == 0)
         {   x1 = -b/(2*a);
             printf("  X1= %f ", x1);
         }
       else
                printf(" The equation does not admit a solution ");
  }
```

# Reminder: The Ternary Operator

In C language the following two instructions are equivalent:

```
if (condition )
        instruction-1      ⇔      condition ? instruction-1 : instruction-2
 else
     instruction-2
```

# Reminder: The Ternary Operator

*Example:*

```
(x >= 0 ) ? x++ : x--
```

This instruction is equivalent to

```
if (x>=0)
        x++
else
        x--
```

# Repetitive control structures (loops)

- A loop is used to repeat statements or  a block of code  a finite number of times following the satisfaction of a condition.
- In C, we distinguish the three loops:
  - The while loop,
  - The do…while loop
  - The for loop,

# The while loop

- This structure repeats a block of instructions as long as the condition mentioned in its header is verified

- Its syntax is as follows :

```
while ( condition )
        {
            instructions
        }
```

# The while  loop

**Example**:

```
#include <stdio.h>
main()
{ int i;
  X = 1;
  while (X < 5)
   {   printf(" X = %d\n",X);
        X++;
    }
}
```

```
X=1
X=2
X=3
X=4
X=5
```

# do…while loop

- ❑ As the condition check is done after the execution of the instruction block, the do…while loop executes at least once even if the condition is not met.

Its syntax is as follows :

```
do {
        Instructions
 } while ( condition )
```

# do…while

Example:

```
#include <stdio.h>
main()
{ int i;
  i = 1;
  do
    {  printf(" i = %d",i);
        i++;
     } while (i >10)
}
```

The program displays

```
i=1
```

# for loop

Its syntax is as follows :

```
for ( initialisation ; condition ; increment/decrement)
   { Instruction
   }
```

❑ An equivalent version using the 'while' loop is:

```
initialisation;
while ( condition )
    {
        instructions
       increment/decrement;
    }
```

# for loop

*Example:*

print all integers from 0 to 10

```
#include <stdio.h>
main()
{  int i;
 for (i=0; i<=10; i++)
 printf(" i = %d",i);
}
```

At the end of this loop, i will be worth 11

# for loop

Example 2: a C program that calculates the square of positive numbers less than 10

```
#include <stdio.h>
#include <math.h>
main()
{  int i;
 for (i=0; i<10; i++)
 printf(" the square  of %d= %d",i, pow (i,2));
}
```