Ministry of Higher Education and Scientific Research
Djilali BOUNAAMA University - Khemis Miliana(UDBKM)
Faculty of Matter Science and Computer Science
Department of Mathematics

Chapter : 3

Conditional Structures (in Algorithmic Language and in C)

MI-L1-UEF121 : Algorithms and Data Structures I

Ali Khalfi

Khalfiali.udbkm@gmail.com

Course Topics

1. Introduction
2. Simple Conditional / Selection statement
3. Compound Conditional / Selection statement
4. Multiple choice conditional statement
5. Branching

1. Introduction

Needs

- ✓ Problems are more complex to be solved with simple instructions.
- ✓ The resolution of certain problems can only be done conditionally.
- ✓ Find an algorithmic structure capable of supporting
  - ➢ the different treatments relating to different conditions
  - ➢ exclusively trigger processing that meets a certain condition.

Block / Statements Block

- ✓ A **block** is a coherent set of one or more primitive actions
- ✓ A **block** begins with a block “Start” : **begin**
- ✓ A **block** ends with an “End” of block : **end**
- ✓ If the **block** is composed of only one (1) action then, “**begin**” and “**end**” of block are optional

Conditional statement

Conditional / Selection Statement

Syntaxe

```
if  (condition) then
        Block 1

    endif
...
```

- ✓ If the **condition** is ***verified*** (true), “Block 1” is executed.
- ✓ If the **condition** is ***not verified*** (false), we move on in sequence after “Block1”
- ✓ The **condition** is a ***logical expression***

Example 1: Solving a 1st degree equation

1st degree equation

write a program to solve the first degree equation: ax+b=0 (we assume that a >0)

Analyze :

With notions of mathematics, The solution of a 1st degree equation is : $x = \frac{-b}{a}$

```
Algorithm equation_1er;

var a, b, x: real ;

begin
    read (a, b);

    if (a <> 0) then
        x ← -b/a;
    endif

    write ('the solution is x = ',x);

end.
```

Simple Conditional Statement

Conditional Statement

Declaration

Examples

Algorithm

Syntax: **IF** condition **THEN** Begin ... End

```
algorithm Example;

var     a, b, x : Real;

begin
    Write('input  a and  b :');
    Read(a, b);

    if (a <> 0) then
      begin
        x := -b/a;
      end;

    Write('the solution is  x = ', x);

end.
```

C

Syntax: **if (**condition**) { ... }**

```
#include <stdio.h>

int main (){

    float a, b, x;
    printf(« Input a and  b : ");
    scanf("%f %f", &a, &b);

    if (a != 0)
    {
        x = -b / a;

    }
    printf(« The solution is x = %f", x);

    return 0;
}
```

3. Alternative statement

Alternative Statement

Syntaxe

```
if  (condition) then
        Block statement 1
else
        Block statement 2
endif
...
```

- ✓ If the **condition** is ***verified*** (true), “Block statement 1” is executed.
- ✓ If the **condition** is ***not verified*** (false), “Block statement 2” is executed.

Example2 : Positive? Negative?

« Positive » or « négative »

Write a program that asks the user for an integer, tests whether that number is positive or not, and displays "**positive**" or "**negative**".

Analyze :

" Request an integer from the user " ➔ read ...

"A is Positive" ➔ A >= 0

"A is negative" ➔ A < 0

```
Algorithm  Pos_Neg;

var  A : integer ;

begin
    write ('Give a number A :');
    read (A);
    if  (A >= 0)  then
          write (A,' is positive');
    else
          write (A,' is negative');
    endif

end.
```

Compound Conditional Statement

Alternative Statement

Declaration

Examples

Algorithm

Syntax: **IF** condition **THEN** Begin ... End
ELSE Begin ... End

```
algorithm Example;

var
        A : Integer;

begin
    Write('Input a number  A :');
    Read(A);

    if (A >= 0) then
        Write(A, ' is positive')
    else
        Write(A, ' is negative')
end.
```

C

Syntax: **if** condition **{ ... } else { ... }**

```
#include <stdio.h>

int main (){
    int a;
    printf(« Input a number  A : ");
    scanf("%d", &a);

    if (a >= 0){
        printf("%d is positive", a);
    }else{
        printf("%d is negative", a);
    }
    return 0;
}
```

Nested Conditional Statement

Syntax

```
if  (condition1) then
        Block statement 1
else
        if  (condition2) then
                Block statement 2
        else {
                Block statement 3
        endif
        ...
endif
...
```

Example3 : Positive? Negative? Null?

« Positive », « négative » or « null »

Write a program that asks the user for an integer, and displays "**strictly positive**", "**strictly negative**", or "**zero**".

Analyze :

"A is strictly positive" ➔ A > 0

"A is strictly negative" ➔ A < 0

"A is null" ➔ A = 0

```
Algorithm Pos_Neg_Nul;

var A : integer ;

begin
    write ('Input a number A :');
    read (A);
    if (A > 0) then
            write (A,' is strictly positive');
    else
            if (A < 0) then
                write (A,' is strictly negative');
            else
                write (A,' is null');
            endif
    endif
end.
```

Compound Conditional Statement

Alternative Statement

Declaration

Examples

Algorithm

Syntax: **IF** condition **THEN** Begin ... End
ELSE Begin ... End

```
algorithm Example;

var
        A : Integer;

begin
    Write('Input a number  A :');
    Read(A);

    if (A > 0) then
        Write(A, ' is strictly positive')
    else
        if (A < 0) then
            Write(A, ' is strictly negative')
        else
            Write(A, ' is null');

end.
```

C

Syntax: **if** condition **{ ... } else { ... }**

```
#include <stdio.h>

int main (){
    int a;
    printf(« Input a number A : ");
    scanf("%d", &a);

    if (a > 0){
        printf("%d is strictly positive', a);
    }else{
        if (a < 0){
            printf("%d is strictly negative', a);
        }else{
            printf("%d is null", a);
        }
    }
    return 0;
}
```

Ladder Conditional Statement

Syntax

```
if  (condition1) then
        Block statement 1
else if  (condition2)
        Block statement 2
else if  (condition3)
        Block statement 3
else if  (condition4)
        Block statement 4
else
        Block statement 5
endif
...
```

Example4 : Positive? Negative? Null?

« Positive », « negative » or « null »

Write a program that asks the user for an integer, and displays "**strictly positive**", "**strictly negative**", or "**zero**".

Analyze :

"A is strictly positive" ➔ A > 0

"A is strictly negative" ➔ A < 0

"A is null" ➔ A = 0

Compound Conditional Statement

Alternative Statement

Declaration

Examples

Algorithm

Syntax: **IF** condition **THEN** Begin ... End
ELSE Begin ... End

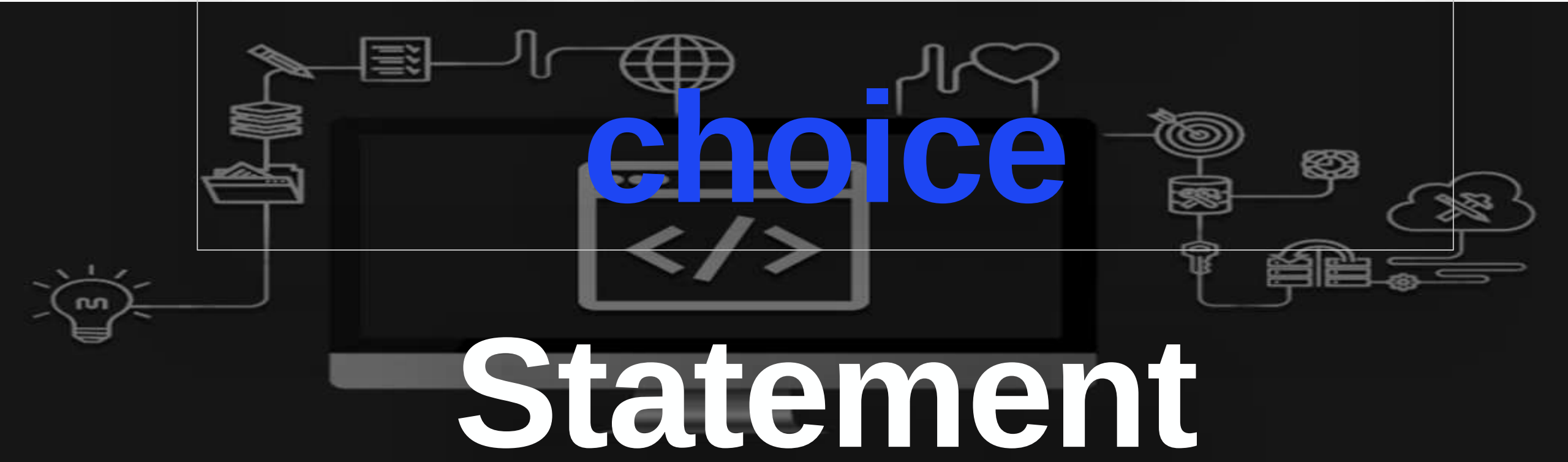
```
algorithm Exemple_Const;

var
        A : Integer;

begin
    Write('Input a number A :');
    ReadLn(A);

    if (A > 0) then
        WriteLn(A, ' is strictly positive')
    else if (A < 0)
        WriteLn(A, ' is strictly negative')
    else
        WriteLn(A, ' is null');

end.
```

C

Syntax: **if** condition **{ ... } else { ... }**

```
#include <stdio.h>

int main (){
    int a;
    printf(« Input a number A : ");
    scanf("%d", &a);

    if (a > 0){
        printf("%d is strictly positive", a);
    }else if (a < 0)
        printf("%d is strictly negative", a);
    }else{
        printf("%d is null", a);
    }
    return 0;
}
```

4. Multiple choice Statement



Multiple choice conditional statement

- ✓ C language offers its users with a **selection statement** in various ways in case a program becomes difficult to read with an increased number of conditions.
- ✓ The **case** declaration comes into play when more than three alternatives (conditions) exist in a program.
- ✓ This command then **casees** between all the available blocks on the basis of the expression value.
- ✓ Then, each block has a corresponding value with it.

case Statement

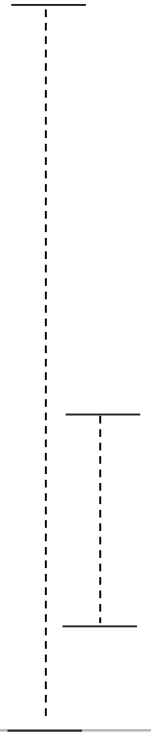
Syntax

```
case (Variable ou Expression) of
        value1 : statement_1;

        value2 : statement_2;
        .
        .
        value_n : statement_n;
else
        default_statement;
endcase
...
```

- ✓ The choice is made according to the value of a **selector** (Variable or expression)
- ✓ This structure allows you to avoid **nested if-else statements**

Example5 : Reservation menu

```
Algorithm  menu;

var  choice : integer ;

begin
    write (' Input your choice  : ');
    write('1. Reserve a  vehicle');
    write('2. Reserve a room');
    write('3. Reserve a flight');

    read (choice);

    case choice of
        1: write('1. Reserve a vehicle');
        2: write('2. Reserve a room');
        3: write('3. Reserve a flight');
        else write('1. Reserve a vehicle');
    endcase

end
```



case Statement

Multiple choice Statement

Declaration

Examples

Algorithm

Syntax: **CASE** variable **OF**

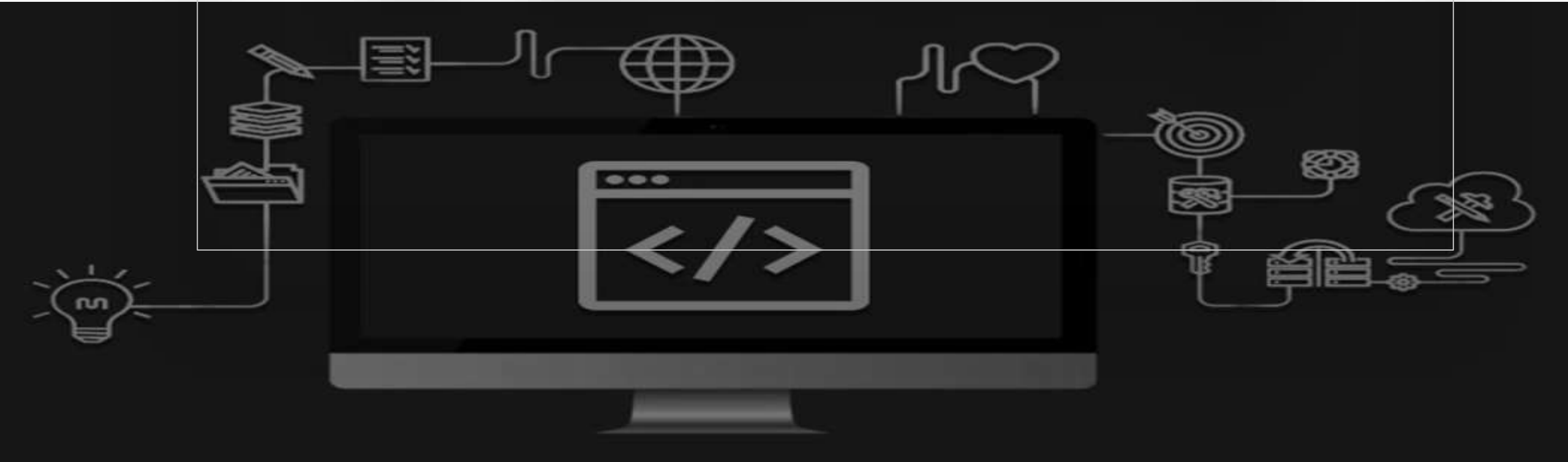
```
algorithm Exemple_Const;

var
        choice : Integer;

begin
    Write('Input your choice  : ');
    Write('1. Reserve a vehicle');
    Write('2. Reserve  a room ');
    Write('3. Reserve a flight ');

    Read(choice);
    Case choice of:
        1: Write('a vehicle is reserved ');
        2: Write('a room is reserved ');
        3: Write('a flight is reserved');
        else Write('Ivalid choice ');
    end;
```

C

Syntax: **case (variable) { case ... }**

```
#include <stdio.h>
int main (){
    int choice;
    printf(« Input your choice  : \n");
    printf("1.  Reserve a vehicle \n");
    printf("2.  Reserve  a room \n");
    printf("3.  Reserve a flight \n");
    scanf("%d", &choice);
    case (choice)
    {
    case 1: printf(" a vehicle is reserved ");
        break;
    case 2: printf(" a room is reserved ");
        break;
    case 3: printf(" a flight is reserved ");
        break;
    default: printf(" Ivalid choice ");
        break;
    }
    return 0;
}
```

5. Branching



goto Statement

Syntax

```
goto Label:
        // block intructions 1
Label:
        // block intructions 2
```

✓ This type of branching doesn't depend on any condition - it always happens.

Unconditional Branching (Jumping)

Declaration

Examples

Algorithm

Syntax: goto

```
algorithm Example;


begin
    Write('Start ');
    goto skip; // Unconditional jump
    Write('This will be skipped  ');
          (Jumping)
    skip:
    printf("After label\n");

end.
```

C

Syntax: goto

```
#include <stdio.h>

int main() {

    printf("Start\n");

    goto skip; // Unconditional jump

    printf("This will be skipped\n");

skip:
    printf("After label\n");
    return 0;
}}
```

Ministry of Higher Education and Scientific Research
Djilali BOUNAAMA University - Khemis Miliana(UDBKM)
Faculty of Matter Science and Computer Science
Department of Mathematics

Chapter : 3

Conditional Structures (in Algorithmic Language and in C)

MI-L1-UEF121 : Algorithms and Data Structures I

Ali Khalfi

Khalfiali.udbkm@gmail.com