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

Chapter 3

Conditional Statements

MI-L1-UEF121 : Algorithms and Data Structures I

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz

Course
Topics

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

Introduction

```
    def __init__(self,
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'),
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in

    @classmethod
    def from_settings(cls, settings):

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

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

```
    def __init__(self, path=None, debug=False):
        self.file = None
        self.fingerprints = set()
        self.logdupes = True
        self.debug = debug
        self.logger = logging.getLogger(__name__)
        if path:
            self.file = open(os.path.join(path, 'requests.seen'), 'a+')
            self.file.seek(0)
            self.fingerprints.update(x.rstrip() for x in self.file)

    @classmethod
    def from_settings(cls, settings):
        debug = settings.getbool('DUPEFILTER_DEBUG')
        return cls(job_dir(settings), debug)

    def request_seen(self, request):
        fp = self.request_fingerprint(request)
        if fp in self.fingerprints:
            return True
        self.fingerprints.add(fp)
        if self.file:
            self.file.write(fp + os.linesep)

    def request_fingerprint(self, request):
        return request_fingerprint(request)
```

Simple Conditional Statement

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***

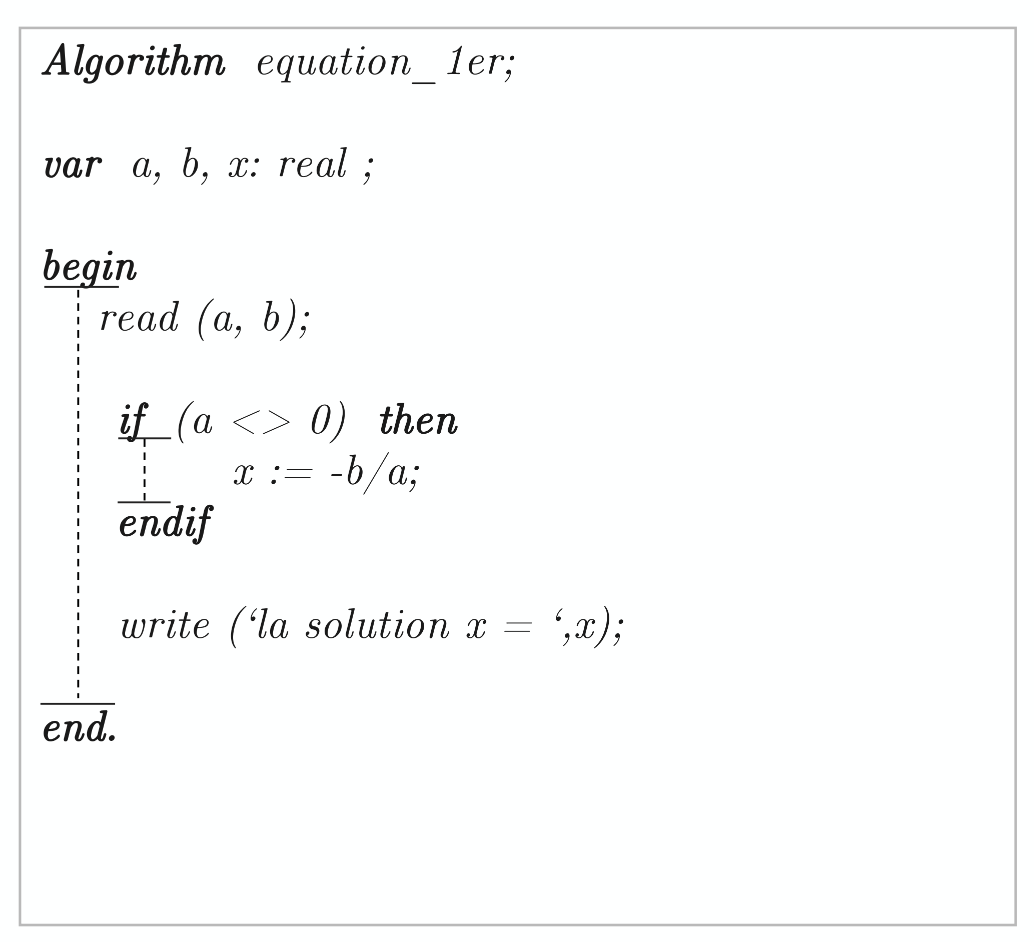
Example 1: Solving a 1st degree equation

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 ('la solution x = ',x);

end.
```

Simple Conditional Statement

Conditional Statement

Declaration

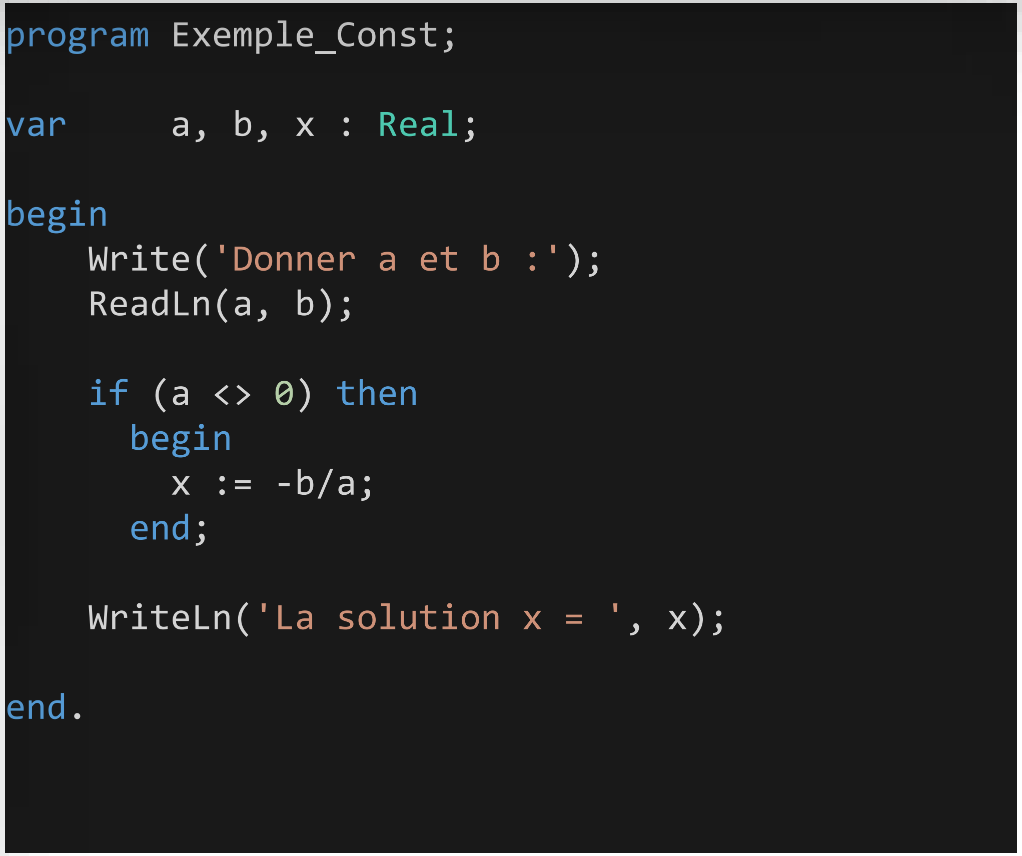
PASCAL

Syntaxe: **IF** condition **THEN** Begin … End

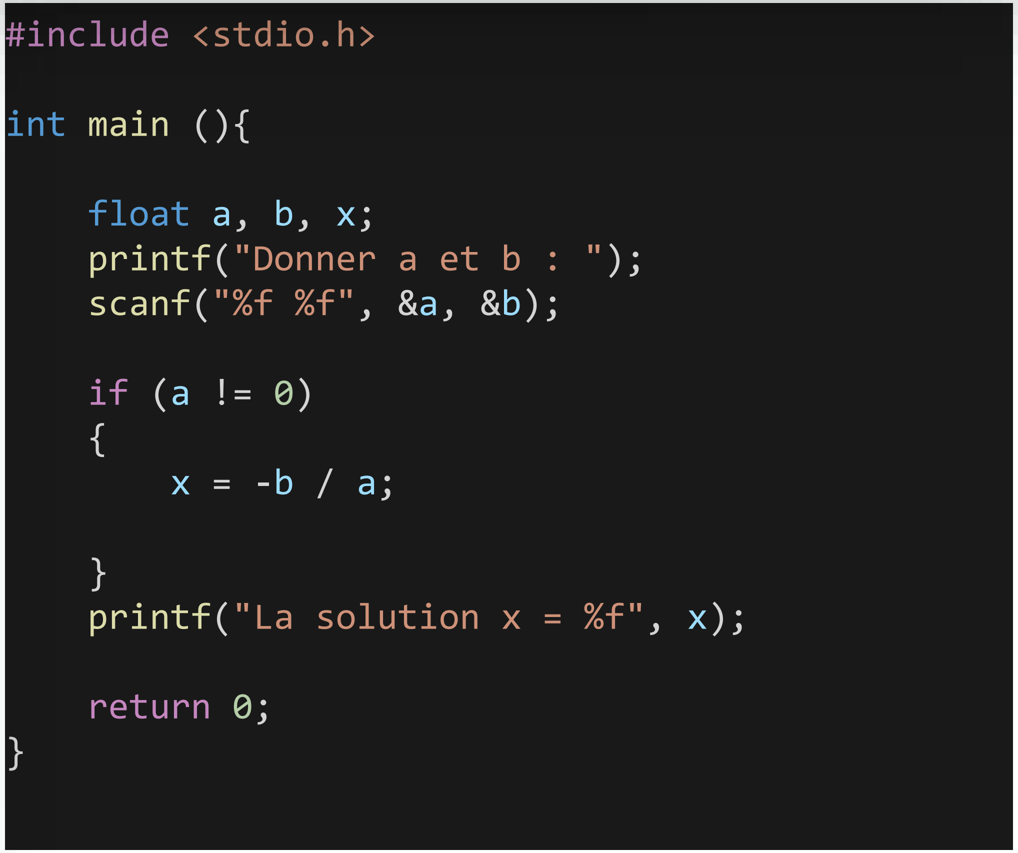
C

Syntaxe: **if (**condition**) { … }**

Exemples



```pascal
program Exemple_Const;

var      a, b, x : Real;

begin
    Write('Donner a et b :');
    ReadLn(a, b);

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

    WriteLn('La solution x = ', x);

end.
```



```c
#include <stdio.h>

int main (){

    float a, b, x;
    printf("Donner a et b : ");
    scanf("%f %f", &a, &b);

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

    }
    printf("La solution x = %f", x);

    return 0;
}
```

Alternative statement

```
def __init__(self,
    self.file = None
    self.fingerprints = set()
    self.logdupes = True
    self.debug = debug
    self.logger = logging.getLogger(__name__)
    if path:
        self.file = open(os.path.join(path, 'requests.seen'),
        self.file.seek(0)
        self.fingerprints.update(x.rstrip() for x in

@classmethod
def from_settings(cls, settings):

def request_seen(self, request):
    fp = self.request_fingerprint(request)
    if fp in self.fingerprints:
        return True
    self.fingerprints.add(fp)
    if self.file:
        self.file.write(fp + os.linesep)

def request_fingerprint(self, request):
    return request_fingerprint(request)
```

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 »

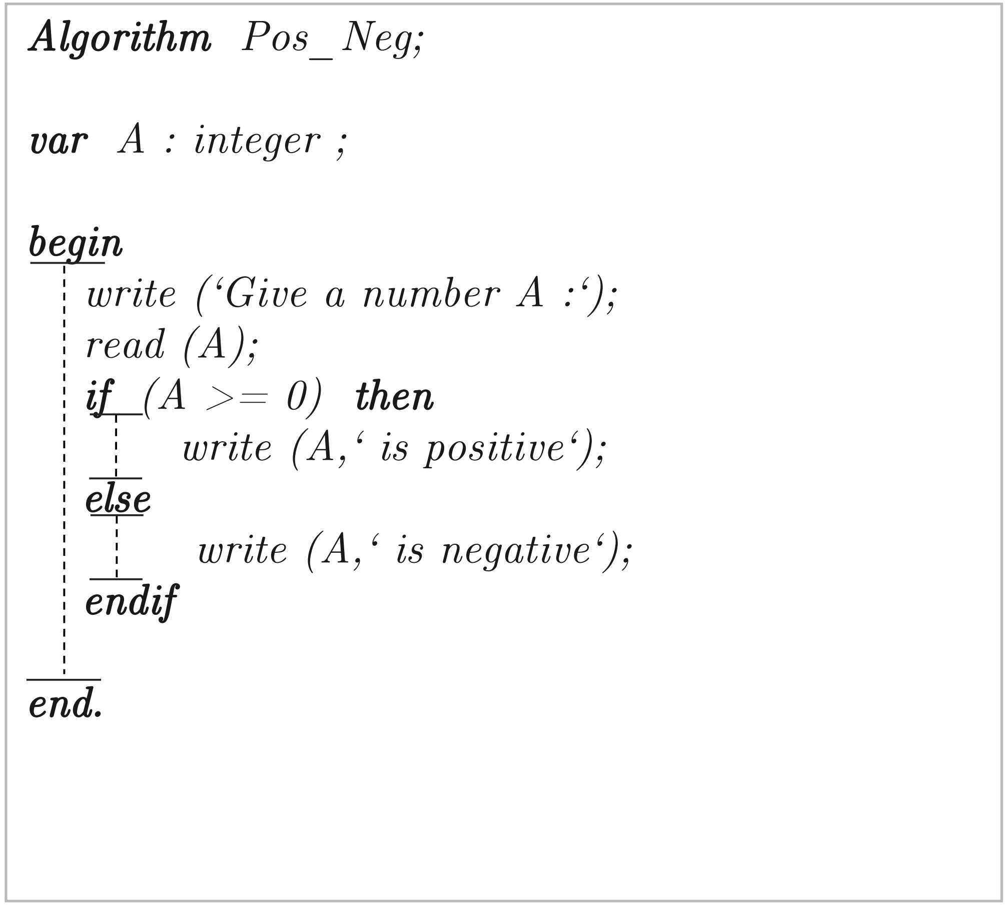
Write a program that asks the user for an integer, tests whether that number is positive or not, and displays "**positive**" or "**negative**".

Analyze :

" Request an integer from the user " ➔ scanf …

"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

PASCAL

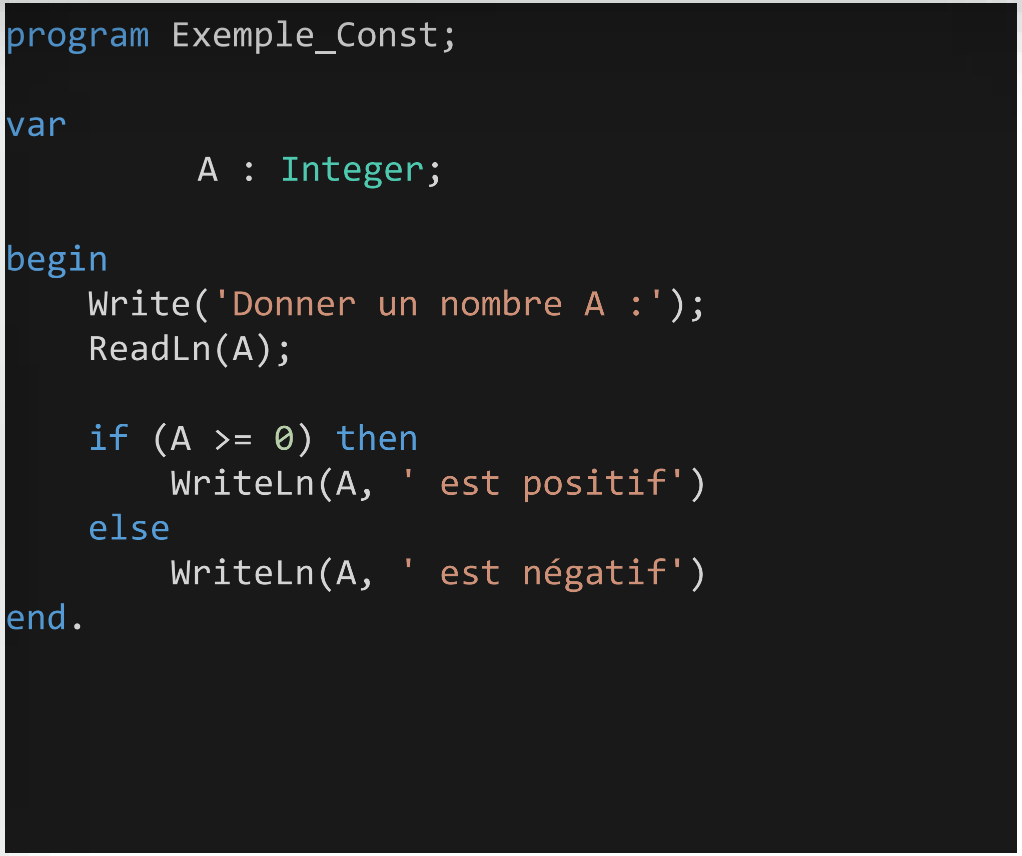
C

Déclaration

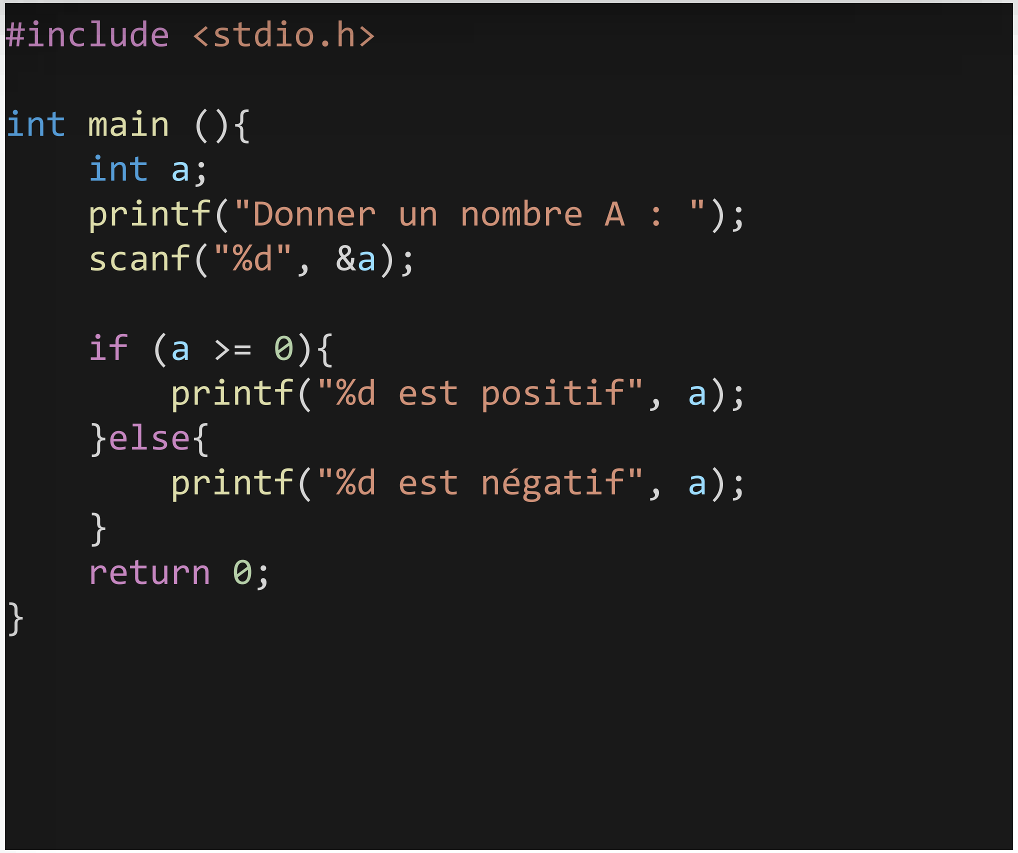
Syntaxe: **IF** condition **THEN** Begin … End **ELSE** Begin … End

Syntaxe: **if** condition **{** … **} else** **{** … **}**

Exemples



```
program Exemple_Const;

var
         A : Integer;

begin
    Write('Donner un nombre A :');
    ReadLn(A);

    if (A >= 0) then
        WriteLn(A, ' est positif')
    else
        WriteLn(A, ' est négatif')
end.
```



```
#include <stdio.h>

int main (){
    int a;
    printf("Donner un nombre A : ");
    scanf("%d", &a);

    if (a >= 0){
        printf("%d est positif", a);
    }else{
        printf("%d est négatif", a);
    }
    return 0;
}
```

Nested Conditional Statement

Syntaxe

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

Compound Conditional Statement

Alternative Statement

Example3 : Positive? Negative? Null?

« Positive », « négative » or « null »

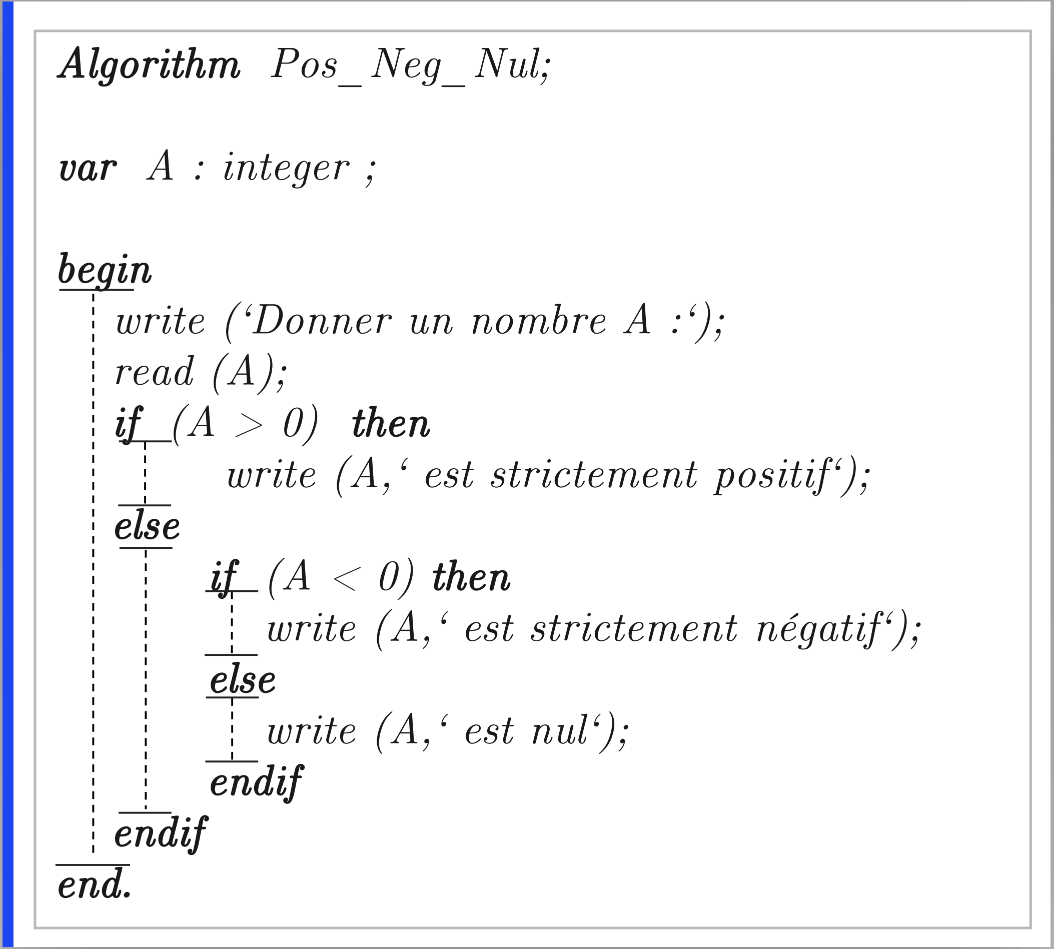
Write a program that asks the user for an integer, and displays "**strictly positive**", "**strictly negative**", or "**zero**".

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 ('Donner un nombre A :');
    read (A);
    if (A > 0) then
        write (A,' est strictement positif');
    else
        if (A < 0) then
            write (A,' est strictement négatif');
        else
            write (A,' est nul');
        endif
    endif
end.
```

Compound Conditional Statement

Alternative Statement

Déclaration

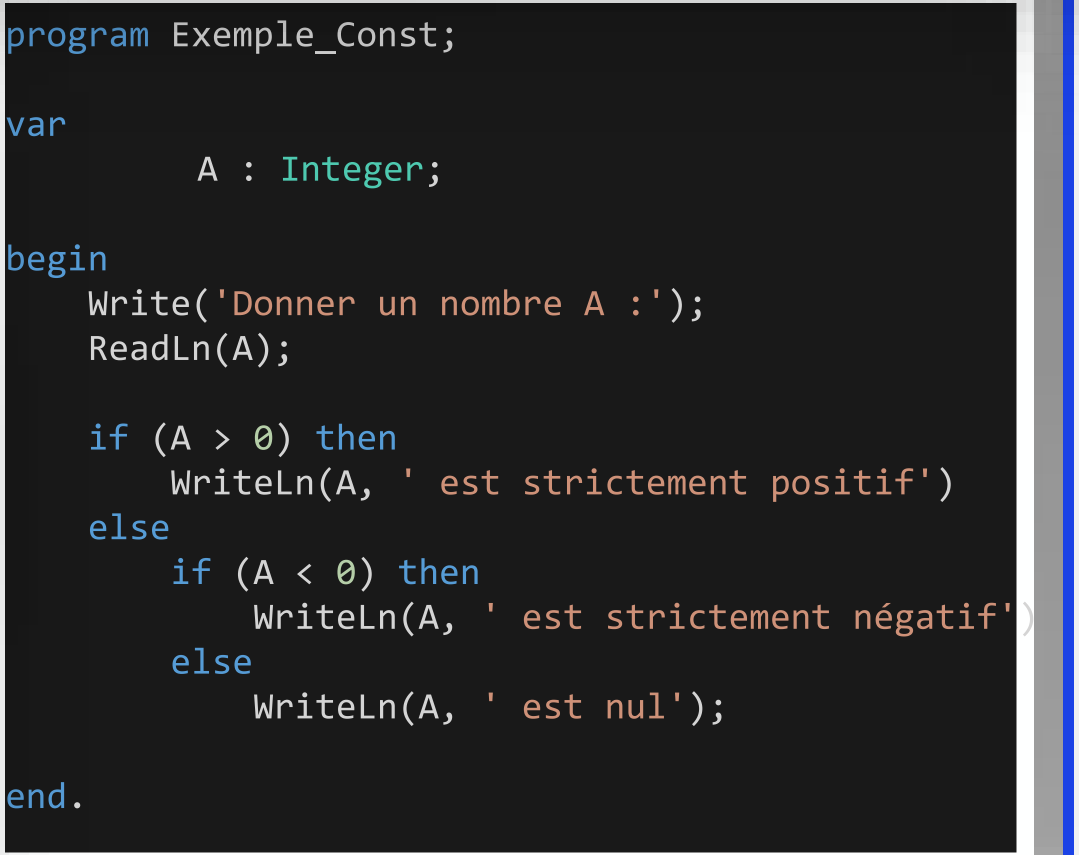
PASCAL

Syntaxe: IF condition THEN Begin … End ELSE Begin … End

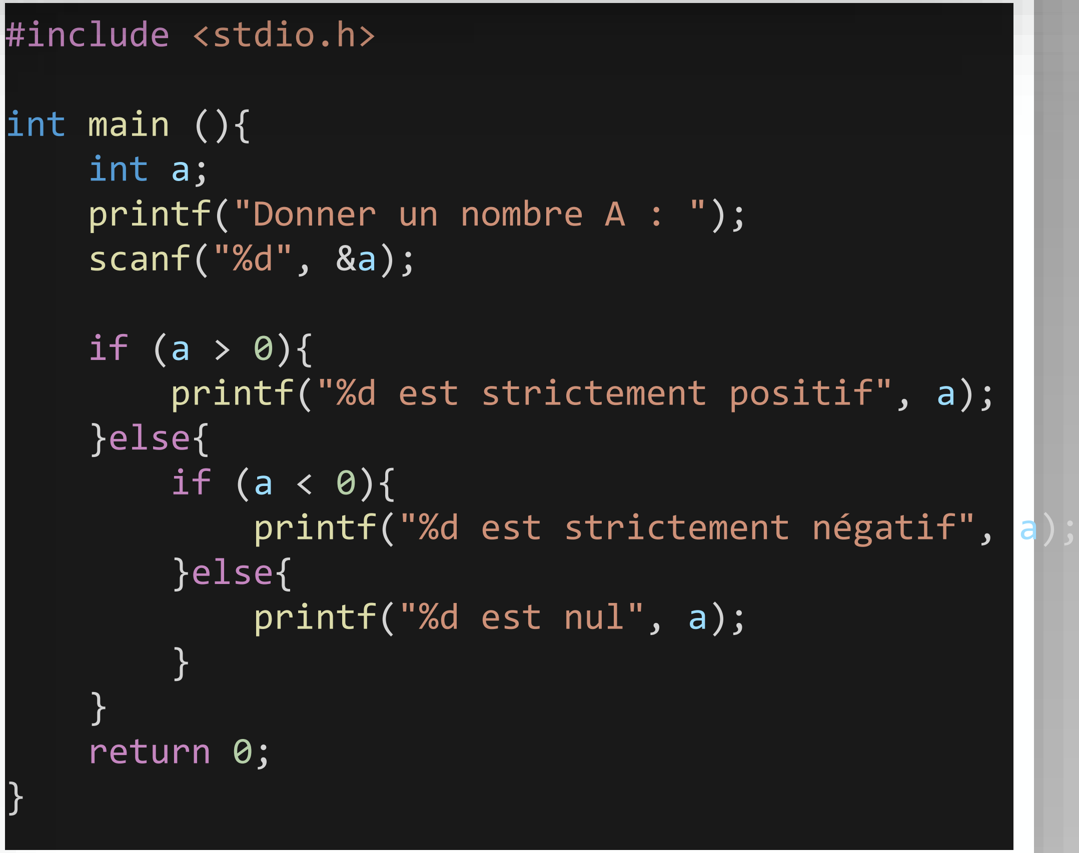
C

Syntaxe: if condition { … } else { … }

Exemples



```pascal
program Exemple_Const;

var
         A : Integer;

begin
    Write('Donner un nombre A :');
    ReadLn(A);

    if (A > 0) then
        WriteLn(A, ' est strictement positif')
    else
        if (A < 0) then
            WriteLn(A, ' est strictement négatif')
        else
            WriteLn(A, ' est nul');

end.
```



```c
#include <stdio.h>

int main (){
    int a;
    printf("Donner un nombre A : ");
    scanf("%d", &a);

    if (a > 0){
        printf("%d est strictement positif", a);
    }else{
        if (a < 0){
            printf("%d est strictement négatif", a);
        }else{
            printf("%d est nul", a);
        }
    }
    return 0;
}
```

Alternative Statement

Ladder Conditional Statement

Syntaxe

```
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
…
```

Compound Conditional Statement

Alternative Statement

Example4 : 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" ➔ $A = 0$

Compound Conditional Statement

Alternative Statement

Déclaration

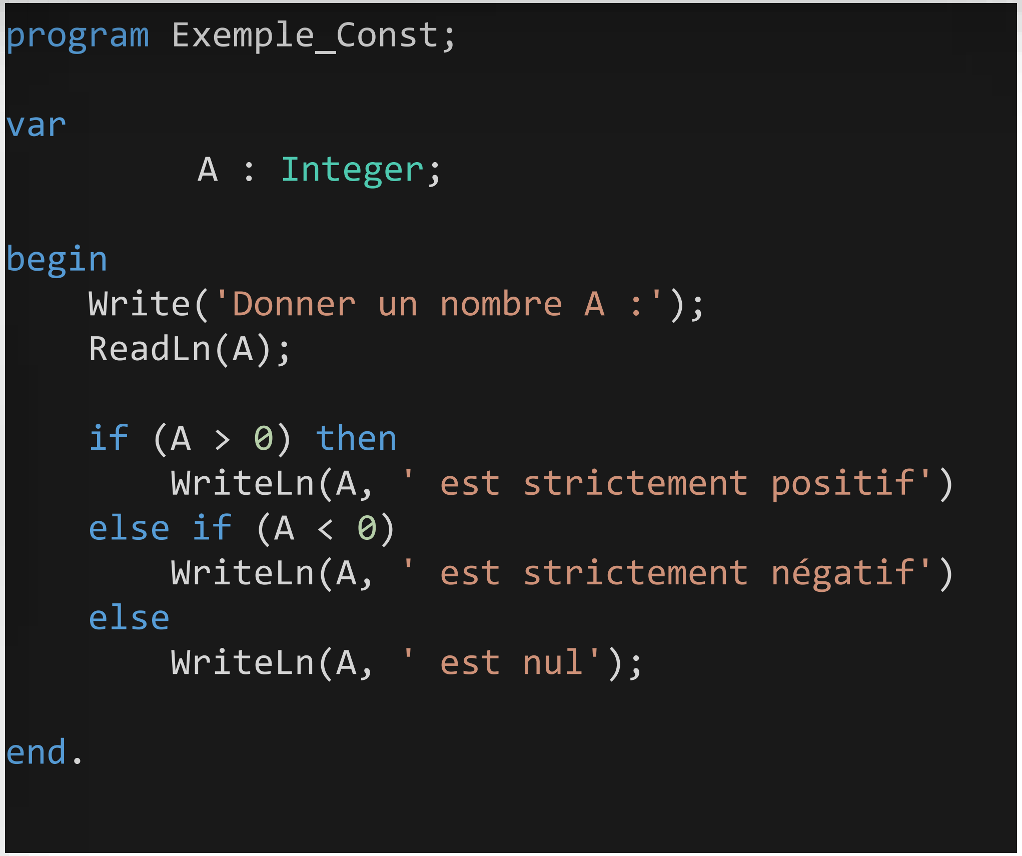
PASCAL

Syntaxe: IF condition THEN Begin … End ELSE Begin … End

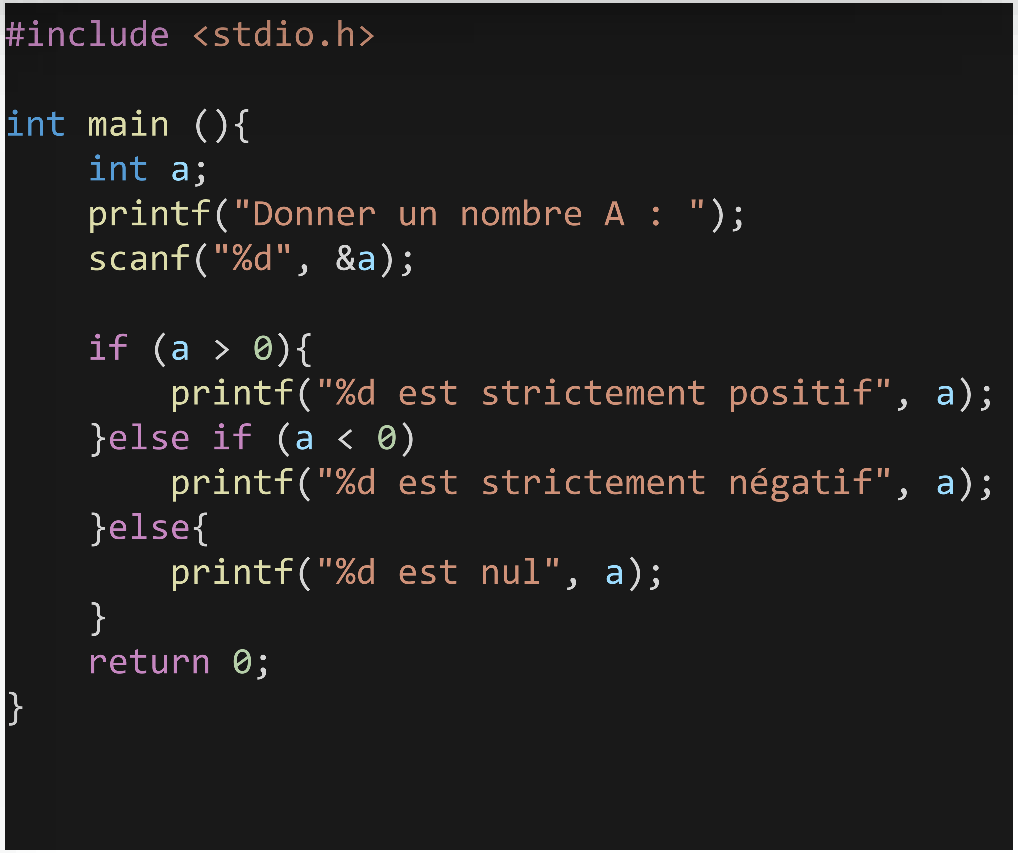
C

Syntaxe: if condition { … } else { … }

Exemples



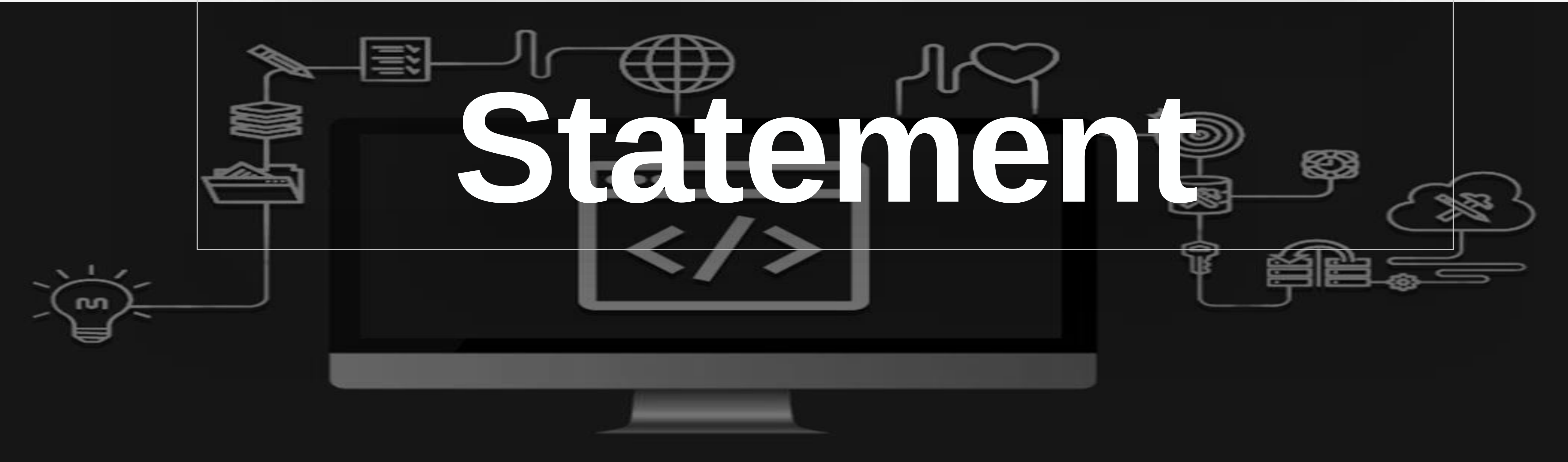
```pascal
program Exemple_Const;

var
         A : Integer;

begin
    Write('Donner un nombre A :');
    ReadLn(A);

    if (A > 0) then
        WriteLn(A, ' est strictement positif')
    else if (A < 0)
        WriteLn(A, ' est strictement négatif')
    else
        WriteLn(A, ' est nul');

end.
```



```c
#include <stdio.h>

int main (){
    int a;
    printf("Donner un nombre A : ");
    scanf("%d", &a);

    if (a > 0){
        printf("%d est strictement positif", a);
    }else if (a < 0)
        printf("%d est strictement négatif", a);
    }else{
        printf("%d est nul", a);
    }
    return 0;
}
```

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.

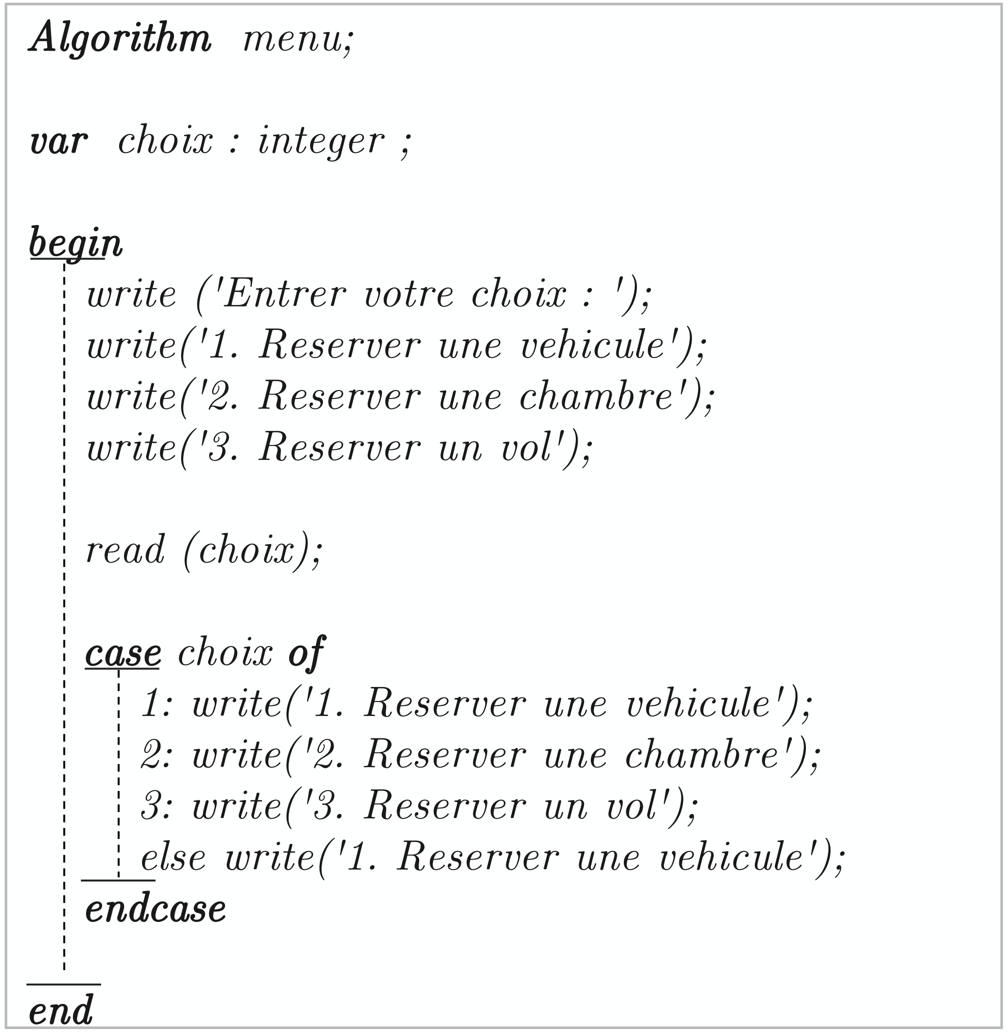
case Statement

Syntaxe

```
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 choix : integer ;

begin
    write ('Entrer votre choix : ');
    write('1. Reserver une vehicule');
    write('2. Reserver une chambre');
    write('3. Reserver un vol');

    read (choix);

    case choix of
        1: write('1. Reserver une vehicule');
        2: write('2. Reserver une chambre');
        3: write('3. Reserver un vol');
        else write('1. Reserver une vehicule');
    endcase

end
```

case Statement

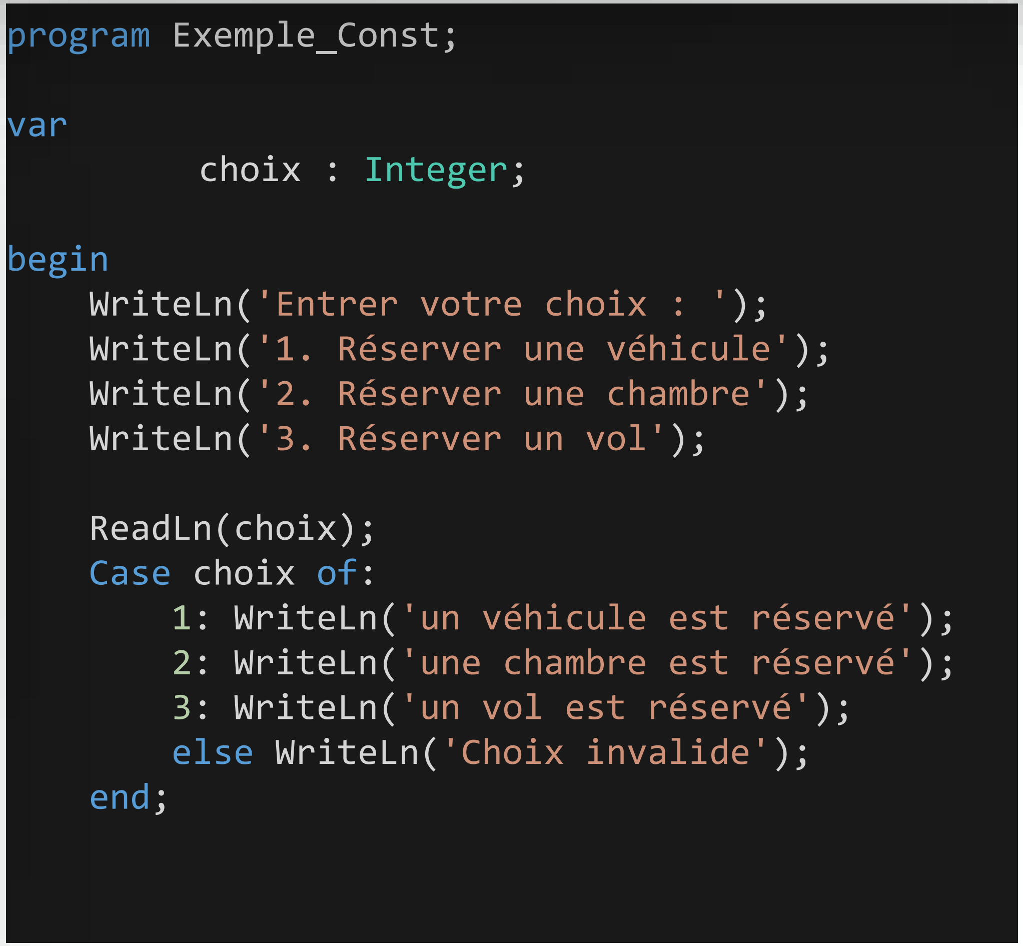
Multiple choice Statement

Déclaration

Exemples

PASCAL

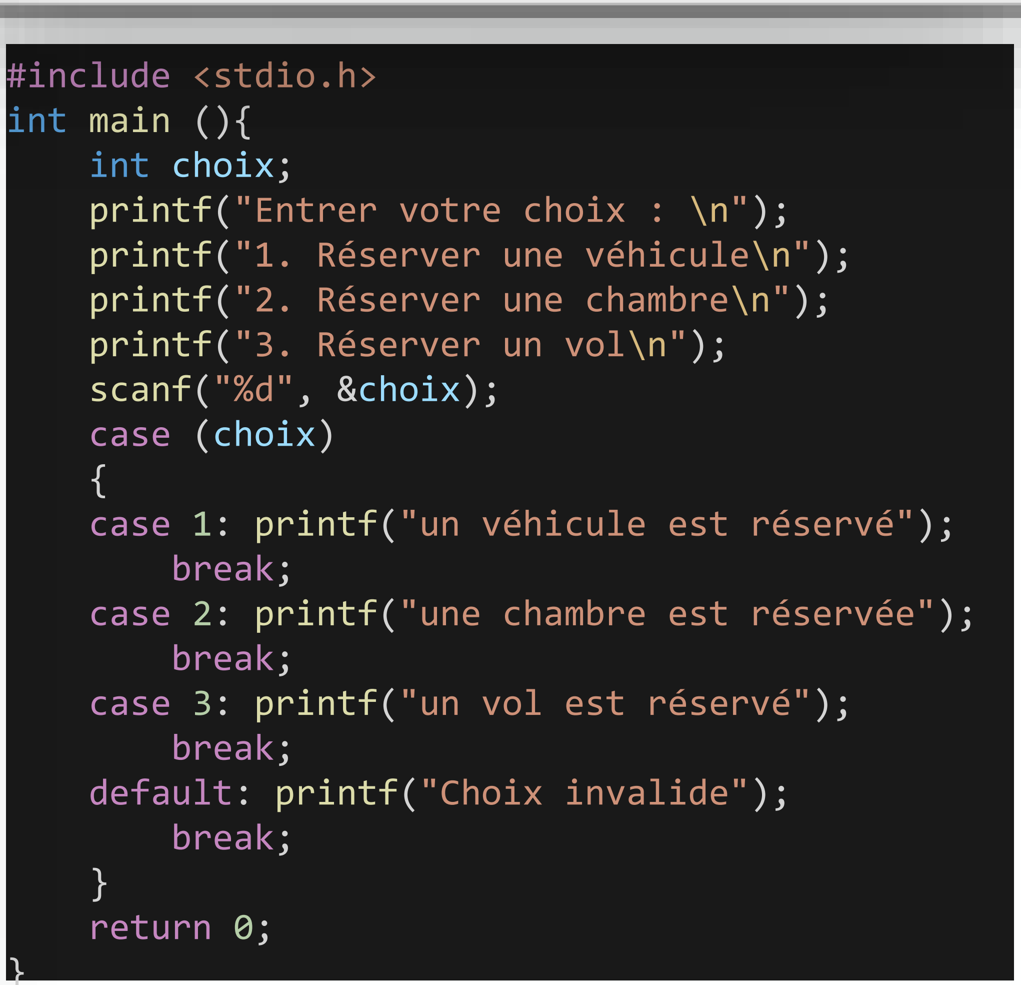
Syntaxe: CASE variable OF



```
program Exemple_Const;

var
        choix : Integer;

begin
    WriteLn('Entrer votre choix : ');
    WriteLn('1. Réserver une véhicule');
    WriteLn('2. Réserver une chambre');
    WriteLn('3. Réserver un vol');

    ReadLn(choix);
    Case choix of:
        1: WriteLn('un véhicule est réservé');
        2: WriteLn('une chambre est réservé');
        3: WriteLn('un vol est réservé');
        else WriteLn('Choix invalide');
    end;
```

C

Syntaxe: case (variable) { case … }



```
#include <stdio.h>
int main (){
    int choix;
    printf("Entrer votre choix : \n");
    printf("1. Réserver une véhicule\n");
    printf("2. Réserver une chambre\n");
    printf("3. Réserver un vol\n");
    scanf("%d", &choix);
    case (choix)
    {
    case 1: printf("un véhicule est réservé");
        break;
    case 2: printf("une chambre est réservée");
        break;
    case 3: printf("un vol est réservé");
        break;
    default: printf("Choix invalide");
        break;
    }
    return 0;
}
```

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

Chapter 3

Conditional Statements

MI-L1-UEF121 : Algorithms and Data Structures I

Noureddine AZZOUZA

n.azzouza@univ-dbkm.dz