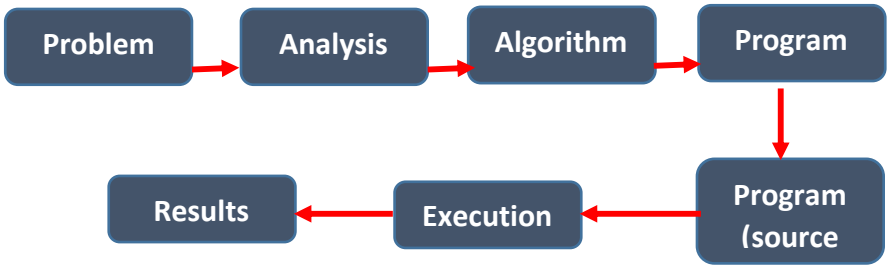
Algorithmic and data structures part I

How to solve a given problem?



Analysis is the step where we clearly understand the problem and determine **what data we have, what result we want, and how to obtain it**.

During analysis, we identify the **variables (inputs)**, the **result (output)**, the **methods or formulas** to use, and the **expressions** that link inputs to outputs.

Definition : **Algorithm :** step by step to solve a problem
Writing the algorithm allows us to organize the solution logically, step by step, independent of any programming language. It helps verify the correctness of the logic before coding.
Synatx
Algorithme name
Variable id: type;
You may also declare:
Arrays;
Const id= value;
Begin
Start of the algorithm.
Instructions
Read(...) input instructions allows user to enter values
Write(...) ouput instructions , allows user to display values or messages
If … Else conditional instructions allows user to use choice according to conditions
Switch
Loops (While, Repeat/Until, For) allows user to repeat actions several times
Assignment allows user to assign values to memory case (variable)
Other operations
End

Assignment
Variable ← expression; (expression can **be** : a value or variable or operation)
Operation (arithmetic or logical operation)
Arithmetic operations: +,-,/,*, div , mod , %, pow,sqrt, sqr, sin …..

Operands must be **numeric**.

Logical operation: Or, and , xor , not (||, &&,!,…), **Operands must be Boolean**
Input /output instructions (entrées / sorties)
Read (variable) ;
Write (‘’message \ n) ; or write (variable) or write (operation) ;

Exercise 01: Write an algorithm that allows the user to enter two reels, then calculate and displays the result of their product
Analysis: so the input data are: two reels, the output is the product so one variable of reel type, the operation is multiplication
Solution:

Alogorithm product
Variable A, B, C: reel ;
Begin
Write (“enter firs reel number \n “); read (A);
Write (“enter second reel number \n “); read (B);
C← A*B;
Write (“ the product of “, A, “and”, B, “is”, C);
End

Exercise 02: Write an algorithm that allows the user to enter the value of **the radius r,** then calculates and displays the **surface of a circle**.

Analysis: the input is radius r, so one variable of reel type, the output is the surface, so one variable of reel type, the method is 3,1’ * radius*radius, so we can declare 3,14 as a const or use the value directly

Algorithm Surface_Circle
Variables
r : real
surface : real
const pi : real ← 3.14
Begin
Write("Enter the radius r : ")
Read(r)
surface ← pi * r * r
Write("The surface of the circle = ", surface); // display message and value of variable
End

Chapter 03: Conditional instructions

If else / switch
Syntax
If(cdt) then instructions
Else instructions
End if

Or If(cdt) then instructions
Else
If(cdt) then instructions
Else
If(cdt) then instructions
Else instructions
End if

Switch (selectors) // The selector is a variable of type int, char, or enum. It determines which case will be executed.
Switch(selector)
Case first_value:
instructions1
Case second_value:
instructions2
Default:
instructions // executed when the value is not defined in any case
EndSwitch

In C language :
switch(selector) {
case value-1:
instructions1;
break;
case value-2:
instructions2;

```
    break;
  default:
    instructions;
}
```

Exercise:

Write an algorithm that displays the maximum of two entered numbers.

Analysis: we have two numbers so two variables de type reel or integer, the result is the max between them, so the method is the operation greater than

```
Algorithme Exo3
Var N, M : entier ;
Begin
    write ("this program display the maximum between two numbers entered by the user \n") ;
    write ("enter the first value\n") ;
    read (N) ;
    write ("enter the second value\n") ;
    read (M) ;
    if (N > M) then
    write ("the maximum is: ", N) ;
    else
    if (M > N) then
    write ("the maximum is : ", M) ;
    else
    write  (‘’ both numbers are equal\n") ;
    end if ;
    endif ;
end.
```

Exercise

write an algorithm and c program that ask the user for two numbers m and n and inform him whether their product is zero or positive or negative.

Analysis:

Inputs: two numbers → variables m and n
Output: the sign of the product (zero, positive, or negative)
Method: analyze the sign of each variable without necessarily computing the product
Expression: sign depends on the combination of signs of m and n

Solution:

```
Algorithm product
Var m, n : integer;
Begin
Write (“give two numbers \n”);
Read(m,n);
If(((n>0) and (m>0)) or ((n<0) and (m<0))) then
Write (“the product is positive\n”);
Else
If (((n>0) and (m<0)) or ((n<0) and (m>0))) then
Write(“the product is negative\”);
Else Write (“the product is zero\n”);
endif
endif
End
```

C program

```
#include <stdio.h>
```

```
int main()
{ int m,n;
   printf("give two integer numbers \n");
   scanf("%d%d",&m,&n);
   if ((n==0)||(m==0))
      printf (" the product is zero\n");
   else
   {
      if (((n>0)&&(m>0))|| ((m<0)&&(n<0)))
         printf (" the product is positive \n");
      else
         printf ("the product is negative \n");
   }
Return 0;
}
```

Chapter 04: Loops

Role: repeat instruction or programs several times
Exercise:
write an algorithm and c program that asks the user for a positive number, then calculates and displays the factorial of this number

```
#include <math.h>
int main()
{    int i,j,fact,n;      float s;
   do
   {printf("give a positive number \n");
   scanf("%d",&n);
   }while (n<0);
   s=1;
  for(i=1;i<=n;i++)
   {  fact=1;
      for(j=1;j<=i;j++)
         fact=fact*j;
      s=s+(pow(i,i)/fact);     }
  printf(" the result is %f",s);
 return 0; }
```

```
Algorithm factorial
Var n ,i,f: integer;
Begin
repeat
write("give a positive number  \n");
read (n);
until(n>=0)
  f ← 1;
for (i from 1 to n step 1 ) do
f= f*i;
end for
write (" the factorial of the number ", n, "is",f);
end.
```

Chapter 05: Arrays and strings

Exercise ask the user to fill an array

```
#include <stdio.h>
int main() {
   float tab[9];   // array of 9 real (float) numbers
   int i;

   printf("Give 9 real numbers:\n");
   // Input loop
   for (i = 0; i < 9; i++) {
      scanf("%f", &tab[i]);
   }
   // Output loop
   printf("You entered:\n");
   for (i = 0; i < 9; i++) {
      printf("%.2f ", tab[i]);
   }
   printf("\n");
```

return 0;

Strings

Specific type of arrays, that contains characters;There are several function in library string.h in C language

Example : write a program that asks the user for a telephone number

Solution

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main() {
   char phone[15];
   int valid, i;
   do {
      valid = 1; // assume the input is valid at the start
      printf("Enter your 10-digit phone number: ");
      scanf("%s", phone);
      // check the length of the input
      if(strlen(phone) != 10) {
         valid = 0;
         printf("Invalid input! Exactly 10 digits are required.\n");
         continue; // go to the next iteration
      }
      // check that each character is a digit
      for(i = 0; i < 10; i++) {
         if(!isdigit(phone[i])) {
            valid = 0; // mark as invalid if any character is not a digit
            printf("Invalid input! Only digits are allowed.\n");
            break; // stop checking further
         }
      }
   } while(!valid); // repeat until the input is valid
   printf("Your phone number is: %s\n", phone);
   return 0; }
```

1. What does the following code display if we enter the two variables chaine1 and chaine2 ?

```
#include <stdio.h>
#include <string.h>
 main ()
{ char chaine1[]= "my name is ";
   char chaine2 [ ]="Mohamed Amine" ;
    gets(chaine1) ;
    scanf("%s",chaine2) ;
    printf("%s\n",strncat(chaine1,chaine2,3)) ;
    printf(strcpy(chaine1,chaine2)) ;
}
```

1. My name is Mohamed amine
   Mohamed amine
2. My name is Mohamed
   Mohamed amine
3. **My name isMoh**
   Mohamed
4. My name isMohMohamed amine