



Tutorial N° 04 : Iterative Instructions

Exercise 01 :

What is displayed on the screen after running these two examples ?

Algorithm Exo1_1

```
Var x : integer ;  
Begin  
    x ← 0 ;  
    While (x <= 5) faire  
        x ← x + 1 ;  
    Write (x) ;  
End.
```

Solution :

X=0, x=0+1, X=1+1=2, X=2+1=3
X=3+1=4 ;x=4+1=5,x=5+1=6 stop

Algorithm Exo1_2

```
Var p, k : integer ;  
Begin  
    p ← 1 ;  
    For k ← 1 à 5 step 1 do  
        p ← p * k ;  
    Write (p) ;  
End.
```

P=1,p=1*1=1;p=1*2=2,p=2*3=6,p=6*4=24;p=24*5=120

Exercise 02 :

Write an algorithm that takes an integer as an input, and then displays the next ten numbers. For example, if the user enters the number 17, the program will display the numbers 18 to 27.

Algorithm exo02

```
Var n,i: integer ;  
Begin  
write ("give an integer number \n");  
read(n);  
for(I from 1 to 10 step 1 )do  
write (n,"+",i,"=",n+i);  
end for;  
end
```

Exercise 03:

Write an algorithm that guarantees that the number entered on the keyboard is positive. If it is negative, you must re-enter another number until the answer matches.

Algorithm exo03

```
Var n : integer;  
Begin  
repeat  
write ("give a positive number \n");  
read(n);  
until (n>0);  
end
```

Exercise 04 :

Write an algorithm that asks the user for a number between 0 and 20 until the answer matches, using the "while" loop first and the "do-while" loop later.

Finally, translate the two solutions into C program.

Algorithm exo04-1

```

Var n : integer;
Begin
repeat
write (“give a number between 0 and 20\n”);
read(n);
until ((n>=0) and (n <=20));
end
    
```

C program

```

#include <stdio.h>
int main()
{ int n ;
  do
  {printf("give a number between 0 and 20 !\n");
   scanf("%d",&n);
  }while ((n<0) || (n>20));
  printf(" correct answer \n");
  return 0;
}
    
```

Algorithm exo04-2

```

Var n : integer;
Begin
write (“give a number between 0 and 20\n”);
read(n);
while((n<0) or (n >20)) do
write (“give a number between 0 and 20\n”);
read(n);
endwhile
end
    
```

```

#include <stdio.h>
int main()
{ int nbr;
  printf("give a number between 0 and 20 !\n");
  scanf("%d",&nbr);
  while ((nbr<0) || (nbr>20))
  {
   printf("give a number between 0 and 20 !\n");
   scanf("%d",&nbr);
  }
  printf(" correct answer \n");
  return 0;
}
    
```

Exercise 05 :

Write an algorithm which requires a number, and which calculates its factorial. The factorial of 8 (noted 8!) is given by 1 x 2 x 3 x 4 x 5 x 6 x 7 x 8

There is no factorial for a negative number

Algorithm factorial

```

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

Exercise 06

When Fatima was born, her grandfather Mohamed opened a bank account for her. Then, on each of her birthdays, Fatima's grandfather paid 1000 AD into her account to which he added double Fatima's age. For example, when she was two years old, he paid her 1004 AD. At the age of 3, he paid her 1006 AD. Write an algorithm that allows you to determine how much money Fatima will have on her n-th birthday.

Algorithm :

```

Var i,age,count: integer;
Begin
repeat
write (“ give the age of Fatima \n”);
read (age);
until (age >0);
count←1000;
for(I from 1 to age step 1 ) do
count ←count +1000+2*i;
end for
write(“ the account of Fatima in her “, age ,”th birthday has “,count,”AD\n”);
end.

```

Exercise 07 :

Consider the following C program. Give its equivalent C program using if-else instruction.

```

#include<stdio.h>
main(){
    int num, numSaved, sumDigits=0;
    do{
        printf("Enter a positive integer less than 10000\n");
        scanf("%d",&num);
    }while(num<0 || num>9999);
    numSaved=num;
    while(num!=0){
        sumDigits=sumDigits+num%10;
        num=num/10; }
    printf("The digits sum of the number %d is %d sumDigits\n",numSaved,sumDigits);
}

```

1- Give the algorithm and the flowchart of this program.

Algorithm exo9

Var num, numsaved,sumdigits:integer;

Begin

sumdigits←0;

repeat

Write (‘enter a positive integer less than 10000 \n’);

read(num);

until((num>=0) and (num <=9999));

numsaved←num;

while (num <>0)**do**

sumdigits←sumdigits +(num mod 10);

num←num div 10;

end while

printf(‘The digits sum of the number ‘, numSaved ,’is “,sumDigits);

end .

2- What loops are used in this program? Can you translate them into the « for » loop ?

Do while and while , the first one , no we can't , the second yes , we can

for(int i=num;i=0;i=i/10)

{ sumDigits=sumDigits+num%10;

num=num/10;

}

3- Translate the « while » loop given in the previous program into « do-while » loop.

```
do{    sumDigits=sumDigits+num%10;
      num=num/10; }
while(num!=0);
```

4- An Armstrong number is a natural number that is equal to the sum of the cubes of its digits. For example, 153 is an Armstrong number because $1^3 + 5^3 + 3^3 = 153$. What changes should be made to the program to check whether an integer is an Armstrong number or not ?

```
while(num!=0){
    sumDigits=sumDigits+pow(num%10,3);
    num=num/10; }
if(numsaved==sumdigits )
    printf("this number is Armstrong number \n");
```

SUP EXER

Exercise 08:

Write an algorithm that calculates the following formula

$$S = 1 + \frac{1}{1!} + \frac{2^2}{2!} + \dots + \frac{n^n}{n!}$$

Algorithm exo 08

```
Var    fact, n, i : entier ;
      s : réel ;
Begin
  Répéter
    Lire(n) ;
  Jusqu'à (n>0) ;
  S ← 1 ;
  Pour i ← 1 à n faire
    Fact ← 1 ;
    Pour j ← 1 à i faire
      fact ← fact * j ;
    end for
    S ← S + pow(i,i) / fact ;
  FinPour ;
  Ecrire(S) ;
End.
```

```
#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;
}
```

Exercise 9 : Display the multiplication tables for integers strictly less than 10.

Algo exo 9

```
Var I, j , n: integer;
Begin
  For (i from 1 to 10 step 1) do
  Write (" multiplication table of ",I, "is\n");
  For (j from 1 to 10 step 1 ) do
    (I,"*",j,"=",i*j);
  Endfor
End for
```

End.

Exercise 10:

What is displayed on the screen after running the following C program when $n=5$.

```
#include <stdio.h>
main() {
    int n, i, j;
    printf("enter a number of lines !\n");
    scanf("%d",&n);
    for(i=1; i<=n; i++) {
        for(j=i; j<=n; j++) printf("*");
        printf("\n"); }
}
```

Exercise 11 :

To multiply a positive integer x by a number y , we can use the following Russian multiplication algorithm: we start from x and y ; at each step, we divide x by 2 (integer division) and multiply y by 2; we stop when x is 0. The result is the sum of the multiples of y corresponding to the odd quotients of x .

Example: $37 * 129 = 4773$

X	y	sum
$37=18*2+1$	129	129
$18=9*2+0$	$129*2=258$	129
$9=4*2+1$	$258*2=516$	$129+516=645$
$4=2*2+0$	$516*2=1032$	645
$2=1*2+0$	$1032*2=2064$	645
$1=0*2+1$	$2064*2=4128$	$645+4128=4773$
0		

Write the algorithm that allows you to perform Russian multiplication.

Algorithm exo11

Var x,y,sum:integer;

Begin

Write('give two integer numbers \n');

Read(x,y);

sum←0;

Repeat

If (x mod 2 =1) then

Sum ←sum+y;

End if

$x \leftarrow x \text{ div } 2$;

$y \leftarrow y * 2$;

until(x=0);

write ('the result of Russian multiplication is ',sum) ;

end .

```
#include <stdio.h>
int main()
{ int x,y,sum=0;
  printf("give two integers \n");
  scanf("%d%d",&x,&y);
  do
  { if(x%2==1)
    sum=sum+y;
    x=x/2;
    y=y*2;
  }while(x!=0);
  printf("the Russian multiplication is %d",sum);
}
```