



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.

Algorithm Exo1_2

Var p, k : integer ;

Begin

p ← 1 ;

For k ← 1 à 5 step 1 do

p ← p * k ;

Write (p) ;

End.

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.

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.

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.

Exercise 05 : Write an algorithm which requires a number, and which calculates its factorial. The factorial of 8 (noted 8!) is given by $1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$. There is no factorial for a negative number.

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.

Exercise 07 : Consider the following C program.

```
#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.
- 2- What loops are used in this program? Can you translate them into the « for » loop ?
- 3- Translate the « while » loop given in the previous program into « do-while » loop.
- 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 ?

Additional Exercises

Exercise 08 :

Write an algorithm that calculates the following formula

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

Exercise 09 :

Display the multiplication tables for integers strictly less than 10.

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.

Examples of solutions

Exercices N°3:

```
#include <stdio.h>
int main() {
    int n, i;
    printf("Give an integer number:\n");
    scanf("%d", &n);

    for(i = 1; i <= 10; i++) {
```

```

        printf("%d + %d = %d\n", n, i, n + i);
    }

    return 0;
}

```

Exercise 04:

```

#include <stdio.h>

int main() {
    int n;

    do {
        printf("Give a number between 0 and 20: ");
        scanf("%d", &n);
    } while (n < 0 || n > 20);

    printf("Correct! The number is between 0 and 20.\n");
    return 0;
}

```

Exercise 06:

```

#include <stdio.h>

int main() {
    int i, age, count;

    // Repeat until a positive age is entered
    do {
        printf("Give the age of Fatima:\n");
        scanf("%d", &age);
    } while (age <= 0);

    count = 1000;

    for (i = 1; i <= age; i++) {
        count = count + 1000 + 2 * i;
    }

    printf("The account of Fatima on her %dth birthday has %d AD.\n", age, count);

    return 0;
}

```