EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA
Subdirección Académica
Departamento de sistemas y computación

Semestre:
Agosto - Diciembre 2021

Carrera:
Ingeniería En Tecnologías De La Información Y Comunicaciones

Materia:
Datos Masivos

Nombre del trabajo:
Práctica 4

UNIDAD A EVALUAR:
Unidad I

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:
Victor Fabian Hernandez Lamarque | 16212536

NOMBRE DEL MAESTRO (A):
JOSE CHRISTIAN ROMERO

Algorithm 1

Define the value of n, this is the fibonacci series, define the function fib1 with a condition, if n is less than 2 it returns the value n and define the value of n, define the function fib1 with a condition, if n is less than 2 returns the value n and fib where int returns an int is equal if n is less than 2, returns 1, otherwise n minus 1, plus, the fib of n minus

```
val n=10
def fib1(n:Int):Int={
    if(n<2)
    {
        return n
    }
    else
    {
        return(fib1(n-1)+fib1(n-2))
    }
}
println(fib1(n))
```

Algorithm 2

We define the fibonacci value in val n, in the variable phi we assign 1 + square root of 5 and we all divide by 2 and in var j we assign the value of var phi squared - (1 - phi) squared and divided by the root square of 5. Define the function of fib2, if n is less than 2, it returns the value of n, because fib 0 = 0 and fib 1 = 1, otherwise it returns the value je prints the final result of the function fib2

```
val n = 8
var phi=((1+math.sqrt(5))/2)
var j=((math.pow(phi,n)-math.pow((1-phi),n))/(math.sqrt(5)))


def fib2(n:Double) : Double ={
if (n<2){
return n
}
else {

    return j
}
}
println(fib2(n))
```

Algorithm 3

defines the function fib3 and receives a value n, three variables are created, initializes the cycle k of 1 to n numbers, c (0) is equal to b (1) plus a (0), result c = 1, a takes the value of b, b takes the value of c and returns the value of a

```
def fib3(n:Int):Int={
var n : Int = 7
var a = 0
var b = 1
var c = 0



    for(k <- 1 to n) {

        c = b + a
        a = b
        b = c
    }
     return a
}
println(fib3(n))
```

Algorithm 4

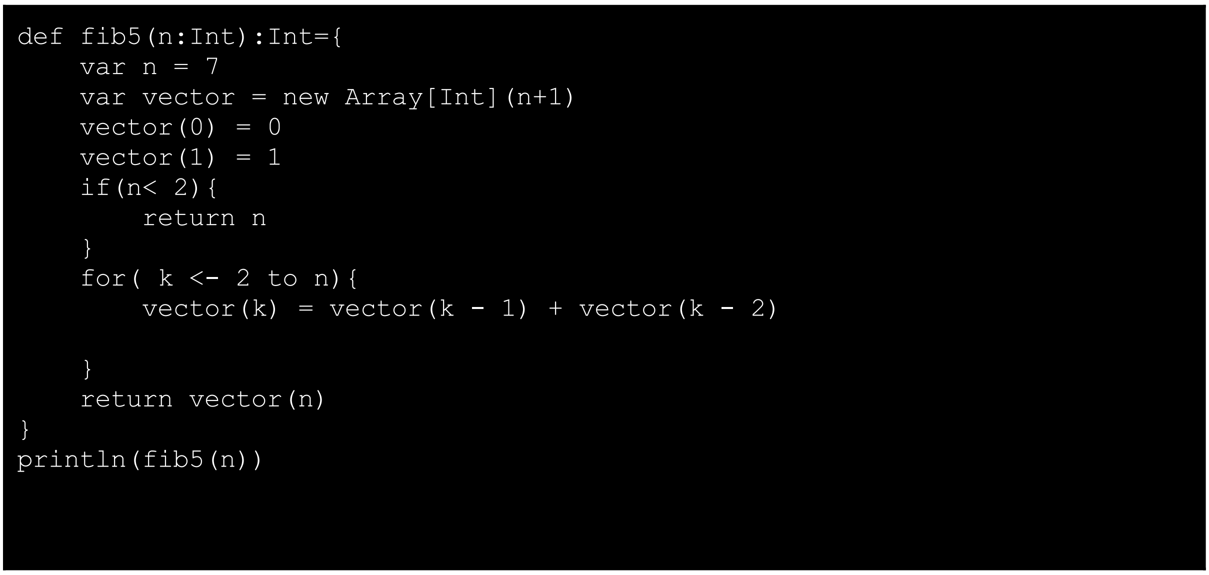
The fib4 function is defined, the number to be used is 6 and 2 variables are defined. The loop is initialized for kan, b is 1, so b is equal to b (1) plus a (0), result 1, a is 0, then a is equal to 1 - 0, result 1 and the loop starts again with new values, in this case b-1 and a-1 and returns a value

```
def fib4(n:Int):Int={
var n : Int = 6
var a = 0
var b = 1

for (k <- 1 to n){
    b = b + a
    a = b - a
}
return a
}
println(fib4(n))
```

Algorithm 5

The fib5 function is defined, n value is defined, we create a matrix of n + 1 positions, we set the position 0 and 1 of the vector to 0 and 1 respectively, since fibonacci from 0 to 0 and 1 to 1, if the value of n is less than 2 returns the same value of n, so the cycle passes in a cycle through vector values, fibonacci operation with vector values with vector values, since the value is 7, the fibonacci value of 6 is add to 8 and the fibonacci value from 5 to 5 and the result is 13 and returns the value of the vector at position n



```
def fib5(n:Int):Int={
    var n = 7
    var vector = new Array[Int](n+1)
    vector(0) = 0
    vector(1) = 1
    if(n< 2){
        return n
    }
    for( k <- 2 to n){
        vector(k) = vector(k - 1) + vector(k - 2)

    }
    return vector(n)
}
println(fib5(n))
```