 

Nombre: María Fernanda Juárez Tirado, 234661

Profesor: jose afredo acuña garcía | **TAREA 3**

PARADIGMAS DE PROGRAMACIÓN

tEMA: PROGRAMACIÓn en prolog

**Universidad Autónoma de Querétaro**

**Facultad de Informática**

Algoritmos de factorial, Fibonacci, Números primos en Python.

Algoritmo Factorial.

Una factorial de n es el producto de todos los números enteros a partir de la 1 a n, inclusivo. La marca de exclamación (!) es el operador unario que representa factorial.

Algoritmo.

*Inicio*

*Definir x, n, num;*

*Escribir la instrucción;*

*Leer el num;*

*N <- 1;*

*X <-1;*

*Si num <- 0, entonces;*

*Escribir un error;*

*Sino*

*Mientras x<-num, hacer*

*N <- n\*x;*

*X <- x+1;*

*Fin Mientras*

*Escribir el resultado*

*Fin*

Código en Python.

*x = int(input("Escribe un número: "))*

*print(x)*

*factorial = 1*

*if x < 0:*

*print("No existe factorial")*

*elif x == 0:*

*print("El factorial es 1")*

*else:*

*for i in range(1,x + 1):*

*factorial = factorial\*i*

*print("El factorial de",x,"es",factorial)*

Algoritmo Fibonacci.

La sucesión de Fibonacci, en ocasiones también conocida como secuencia de Fibonacci, es en sí una sucesión matemática infinita. Consta de una serie de números naturales que se suman de a 2, a partir de 0 y 1. Básicamente, la sucesión de Fibonacci se realiza sumando siempre los últimos 2 números (todos los números presentes en la sucesión se llaman números de Fibonacci).

Algoritmo.

*Inicio*

*Declarar k*

*Función fibo(n)*

*Variable i*

*Fib : arreglo de 1 a n*

*Si*

*Fib[1]=1;*

*Fib[2]=1;*

*Si i = 3 a n de*

*Fib[i]=fib[i-1] + fib[i-2];*

*Fibonacci = fib[n];*

*Fin*

*Si*

*Imprimir k*

*Salida, Fibonacci ((k+2) – 1)*

*Fin*

Código en Python.

*def fibonacci(contador,n,p1,p2):*

*var = ""*

*if(contador!=n):*

*var=fibonacci(contador+1,n,p2,p1+p2)*

*var=str(p2)+" "+var*

*return var*

*n = int(input("Ingrese un numero entero\n"))*

*if(n>0):*

*a=fibonacci(0,(n-1),0,1)*

*print ("0 "+a)*

Algoritmo Números primos.

El algoritmo más simple, para determinar si un número es primo o compuesto, es hacer una serie de divisiones sucesivas del número, con todos los números menores que él, si alguna división da como residuo 0 o es divisible con el número entonces es compuesto en caso contrario es primo.

* Primero verificar si es divisible con 2,3 y 5. Si es divisible entonces es compuesto.
* Segundo hallar la raíz cuadrada del número, si la raíz es exacta entonces el número es compuesto.
* Si la raíz es inexacta, entonces se prueba si es divisible con todos los números primos menores que la raíz obtenida, excepto 2, 3 y 5.
* Si es divisible con algún número primo entonces es compuesto, en caso contrario es primo.

Código en Python.

*MAX = 100*

*primos = [2]*

*for k in range(3, MAX):*

*for i in range(len(primos)):*

*if k%primos[i]==0: break*

*if i==len(primos)-1: #solo si no es múltiplo de los actuales primos lo añade*

*primos.append(k)*

*print (primos);*

Bibliografía.

* <http://pier.guillen.com.mx/algorithms/05-aritmetica/05.6-fibonacci.htm>
* <http://diagramas-de-flujo.blogspot.mx/2013/01/Fibonacci-Recursivo-en-Python.html>
* <http://python-esp.blogspot.mx/2010/02/calcular-numeros-primos.html>