

INGENIERÍA MECATRÓNICA



DI_CERO

DIEGO CERVANTES RODRÍGUEZ

INSTRUMENTACIÓN VIRTUAL

PYTHON 3.9.7, C# & LABVIEW

Factorial de un Número n !

Contenido

Instrucciones – Factorial de un Número n:	2
Código Python – Visual Studio Code (Logo Azul):	2
Resultado del Código Python	3
Código C# (.Net Framework) – Visual Studio (Logo Morado):	3
Código con Resultado en Consola:	3
Resultado del Código C#	5
Diagrama LabVIEW:	5



Instrucciones – Factorial de un Número n:

Calcule el factorial de un número n ($n < 2$).

Pseudocódigo.

1. Lectura del número entero positivo n.
2. Obtener la relación $n! = 1 * 2 * 3 * 4 * 5 * \dots * (n - 1) * n$, por medio de un ciclo. Para esto se hace lo siguiente:

```
factn = 1  
  
for (int i = 1; i < n+1; i++) {  
    factn = factn * i;  
}
```

3. Mostrar la salida en pantalla.

Código Python – Visual Studio Code (Logo Azul):

```
# -*- coding: utf-8 -*-  
  
#Comentario de una sola linea con el simbolo #, en Python para nada se deben poner acentos sino el programa  
#puede fallar o imprimir raro en consola, la siguiente línea de código es para que no tenga error, pero aún  
#así al poner un acento, la palabra saldrá rara en consola.  
#La línea debe ponerse tal cual como aparece y justo al inicio.  
  
#Con cls se borra el contenido de la consola y con el botón superior izquierdo de Play se corre el programa.  
  
#CÓDIGO PARA INTRODUCIR EN CONSOLA UN NÚMERO N Y SACAR SU FACTORIAL:  
  
#Lectura de coeficientes a, b y c de la fórmula cuadrática:  $ax^2 + bx + c = 0$   
#El método input() sirve para imprimir en consola un mensaje y luego permitir al usuario ingresar un dato por  
#consola que será de tipo String.  
#Se usa el método int() para convertir el tipo primitivo del input (que viene siendo un String) a tipo primitivo  
#int y se guarde en una variable, se usa el tipo primitivo int porque es el que manejan los bucles for en sus  
#condiciones  
n = int(input("Ingresa un número entero positivo n: \n"))#n para sacarle su factorial  
  
#Esta es la base del número al cual le sacaremos su factorial, esto lo haremos a continuación a través de un  
#condicional if o un bucle for  
factorial = 1  
  
#La fórmula del factoria es:  $n! = 1*2*3*\dots*(n-1)*n$ 
```

```
#Pero cuando el número n es 0 o 1, el resultado es siempre 1, por eso es que se crea un condicional
#CONDICIONAL IF: En Python este no utiliza llaves de apertura o cierre, solamente se utilizan dos puntos para indicar
#el inicio del condicional, ya sea para el if, else if o else.
if(n == 0 or n == 1):
    factorial = 1
else:
    #BUCLE FOR: En Python este no utiliza llaves de apertura o cierre tampoco, solamente se utilizan dos puntos para
    #indicar el inicio del bucle. Además después de la palabra reservada "for" se declara una variable local que solo
    #se usará en el bucle y será la que cuente hasta que este llegue al extremo indicado dentro del paréntesis de la
    #palabra reservada range() para terminar el bucle, dentro del paréntesis de la palabra reservada "in range()" se
    #coloca el inicio del conteo y el final para indicar cuantas veces se ejecutará el bucle.
    for i in range(2, n+1):
        factorial = factorial*i
    #En este caso como en condicional anterior se analizó la opción para cuando n = 0 o 1, n en el bucle va desde+
    #2 hasta n+1 porque el último número del conteo en el for nunca lo va a alcanzar y se busca que el bucle llegue
    #hasta n.

#Impresión en pantalla con concatenación de varios valores: Cuando se quiere concatenar más de un valor lo que se hace
#es separar entre comillas el mensaje que se quiere imprimir y el valor de las variables que se quiere concatenar
print("El factorial del número ", n, " es igual a : ", factorial)
```

Resultado del Código Python

```
PROBLEMAS  SALIDA  TERMINAL  CONSOLA DE DEPURACIÓN

PS C:\Users\diego\OneDrive\Documents\Aprendiendo\Python\1.-Instrumentación Virtual> & D:/Users/diego/AppData/Local/Programs/Python/Python39/python.exe
"c:/Users/diego/OneDrive/Documents/Aprendiendo/Python/1.-Instrumentación Virtual/3.-Factorial de un Número n.py"
Ingresa un número entero positivo n:
5
El factorial del número 5 es igual a : 120
PS C:\Users\diego\OneDrive\Documents\Aprendiendo\Python\1.-Instrumentación Virtual> █
```

Código C# (.Net Framework) – Visual Studio (Logo Morado):

Código con Resultado en Consola:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace _3._Factorial_de_un_Número_n
{
    class Factorial
    {
        //El código en C# se corre presionando CTRL + F5
        static void Main(string[] args)
        {
            int n; //Número introducido para sacar su factorial
            //Método WriteLine de la clase Console para imprimir en consola y después ejecutar un salto de línea
            Console.WriteLine("Ingresa un número entero positivo n:");

            /*El método ToInt16 de la clase Convert lo que hace en este caso es recibir un String proveniente
            de lo que el usuario ingrese en consola al usar el método Console.ReadLine() y lo convierte a un
            número entero con signo de 16 bits, que abarca los números de -32768 a 32767.*/
            n = Convert.ToInt16(Console.ReadLine());
```

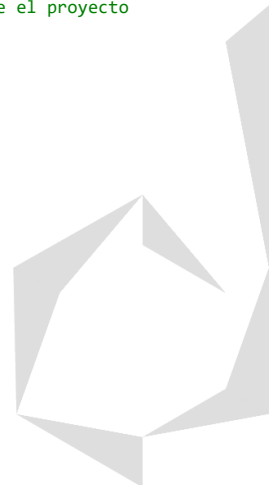
```

/*Se declara una variable de tipo primitivo unsigned long (ulong), que abarca los números de
0 a 18,446,744,073,709,551,615 esto solo permitirá obtener hasta el factorial de un número n < 20*/
//ulong factorial = 1;
/*Si se quiere obtener el factorial para números mayores a 20, se puede usar un tipo de dato Double
el tema es que cuando se haga esto el resultado aparecerá con notación científica y en consola la coma
será equivalente al punto, por lo que no saldrá el número completo*/
double factorial = 1;

//GESTIÓN DE EXCEPCIONES
/*TRY CATCH: Es una forma de decirle al código que debe hacer cuando ocurra un error, para ello se
puede utilizar el try para indicar que debe hacer el código cuando no ocurra error y en el catch
se declara lo que debe hacer el programa cuando ocurra un error.*/
try
{
    /*CONDICIONAL IF: Analiza una condición por medio de operadores lógicos, si el resultado de la condición
es true se ejecuta el código que se encuentra dentro de las llaves del if, sino se brinca al else o si
este no existe, simplemente se brinca todas las líneas de código que ocupa el condicional, los distintos
operadores lógicos que se pueden usar son:
- Igual a (a == b): Este lo que hace es analizar la igualdad entre dos elementos a y b
- OR (a || b): Con el operador lógico OR con que solo una de los valores booleanos sea true, el resultado
lo será, es más utilizado para analizar en conjunto dos operaciones lógicas a y b y comparar sus
resultados.*/
    if ((n == 0) || (n == 1))
    {
        factorial = 1; //Cuando n = 0 o 1, su factorial vale 1.
    }
    else
    {
        //La fórmula del factoria es: n! = 1*2*3*...*(n-1)*n
        /*BUCLE FOR: El bucle for en C# se declara como en la mayoría de los lenguajes de programación,
indicando una variable local de tipo primitivo número entero, hasta donde va a llegar el conteo y
el paso con el que irá contando.*/
        for (int i = 1; i < n + 1; i++)
        {
            /*El método ToUInt64 de la clase Convert lo que hace en este caso es recibir un número entero
y lo convierte en un número entero con signo de 64 bits, que abarca los números de
-9223372036854775808 a 9223372036854775807, se debe hacer la conversión porque un dato ulong o
double no se puede multiplicar por un tipo de dato entero, pero es necesario declararlo así en
la condición del bucle for, esto se usa cuando la variable factorial es de tipo ulong y muestra
el valor del número completo pero solo puede sacar el factorial de hasta 20, se convierte a
double con el método ToDouble de la clase Convert cuando la variable factorial también fue
declarada como double arriba, pero el resultado en este caso aparecerá con notación científica*/
            factorial = factorial * Convert.ToDouble(i);
        }
        /*Cuando se quiere contatenar valores en C#, lo que se hace es usar el método Console.WriteLine y
luego en donde se quiera colocar el valor de la variable se pone su índice, este depende del orden en
el que se hayan declarado los parámetros puestos en el método después del mensaje impreso en consola,
se indica de la siguiente manera: {índice}*/
        Console.WriteLine("El factorial del número {0} es igual a : {1}", n, factorial);
    }
}
/*Lo que se hace en este caso del catch en la gestión de errores es recopilar el error que haya ocurrido
en el programa por medio del parámetro de la operación catch que pertenece a la clase Exception y lo imprime
tal cual en consola*/
catch (Exception error_ups)
{
    Console.WriteLine(error_ups); //Imprimir en consola el error que haya ocurrido
}

} //Método main: Desde el método main se ejecutan todas las partes del proyecto
} //Clase del proyecto
} //Espacio de nombres: En esta parte del código se pueden declarar más de una clase que conforme el proyecto

```



Resultado del Código C#

```
C:\WINDOWS\system32\cmd.  X  +  v

Ingresa un número entero positivo n:
5
El factorial del número 5 es igual a : 120
Presione una tecla para continuar . . . |
```

Diagrama LabVIEW:

