

Trabajo Práctico 1: Introducción

Ejercicio 1

Primer programa en Python. Escribir un programa que imprima por pantalla: “Hola mundo”.

Ejercicio 2

Escribir un programa que cargue en una variable un número entero e imprima por pantalla su tabla de multiplicar (del 1 al 12).

Ejercicio 3

Escribir un programa que cree dos variables enteras y muestre por pantalla, la suma, la resta, la multiplicación, la división entera y el resto de la división entera.

Ejercicio 4

Escribir un programa que realice la transformación de grados Celsius a Fahrenheit, para el valor 20 C y lo imprima por pantalla.

Ejercicio 5

Implementar algoritmos que permitan:

- a) Calcular el perímetro y el área de un rectángulo, dada su base y su altura.
- b) Calcular el perímetro y el área de un círculo dado su radio.
- c) Calcular el volumen y el área de una esfera dado su radio.
- d) Dados los catetos de un triángulo rectángulo, calcular la hipotenusa y el ángulo (expresado en grados).

Nota: Utilizar el paquete math.

Ejercicio 6

Declarar dos variables enteras (con cualquier valor) e informar por pantalla cual es menor de las dos, si son iguales, indicarlo por separado. Cambiar el orden de los valores para comprobar el funcionamiento.

Ejercicio 7

Escribir un programa que declare un numero entero con cualquier valor e indique si dicho número es par o impar.

Ejercicio 8

Escribir un programa que declare un numero entero del 1 al 7 y muestre por pantalla el dia de la semana correspondiente. Controlar que el numero se encuentre en el rango correcto, si no es asi, informar un error. Si el numero es 2 el dia es martes.

Ejercicio 9

Mostrar por pantalla todos los numeros enteros entre 1 y 100, hacerlo con usando un bucle *while* y tambien con un bucle *for*.

Ejercicio 10

Mostrar por pantalla todos los numeros enteros entre 1 y 100 divisibles por dos y por tres.

Ejercicio 11

Escribir un programa que imprima por pantalla todas las fichas del domino, una por linea, sin repetir.

Ejercicio 12

Escribir un programa que pida un numero natural n al usuario e imprima por pantalla los n primeros numeros triangulares y su indice. El numero triangular de indice n se define como la suma de los numeros naturales de 1 hasta n . Por ejemplo para $n = 5$, la salida debe ser:

1 – 1

2 – 3

3 – 6

4 – 10

5 – 15

$$T(n) = \sum_{i=1}^n i = \frac{n(n+1)}{2}$$

Nota: Hacerlo de dos formas, usando y sin usar la ecuacion:

Ejercicio 13

Construir un programa que lea un número natural N y calcule la suma de los primeros N números pares.

Ejercicio 14

Escribir un programa que recibe como entrada desde el usuario los valores de a , b y c de un polinomio de grado 2, e informa las raices (reales y complejas).

Ejercicio 15

Escribir un programa que recibe como entrada desde el usuario dos numeros enteros e informa por pantalla todos los numeros pares entre ellos.

Ejercicio 16

Escribir un programa que pida un numero entero y muestre por pantalla la cantidad de cifras de dicho numero.

Ejercicio 17

Escribir un programa que pida un numero entero e informe si dicho numero es primo o no primo. Nota: Ningun numero es divisible por un numero mayor a la raiz cuadrada de si mismo.

Ejercicio 18

Escribir un programa que imprima por pantalla los numeros desde el 32 al 64, en todas las bases desde 2 hasta 16. Se debe imprimir un numero por fila en las 15 bases distintas separados por el carácter tabular (“\t”). Nota: Los numeros se pueden imprimir en orden inverso.

Ejercicio 19

Escribir un programa que calcule el factorial de un numero natural introducido por el usuario. La operación factorial (!) se define de la siguiente manera:

$$N! = \begin{cases} 1 & \text{si } N=0 \\ N(N-1)! & \text{si } N>0 \end{cases} = N(N-1)(N-2)...3.2.1$$

Ejercicio 20

Escribir un programa que calcule el numero combinatorio de dos numeros naturales m y n . El numero combinatorio de m agrupados de a n se calcula de la siguiente manera:

$$C = \binom{m}{n} = \frac{m!}{n!(m-n)!}$$

Implementar dentro del programa, una funcion que calcule el factorial de un numero.

Nota: Se debe controlar que ambos numeros sean naturales y mostrar por pantalla un error en caso contrario.

Ejercicio 21

La funcion $f(x) = e^x$ se puede aproximar con una determinada tolerancia de error (TOL) a traves de un polinomio de Taylor de grado N , usando a siguiente formula:

$$e^x \approx \sum_{i=1}^N \frac{x^i}{i!} = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots + \frac{x^N}{N!}$$

El valor de N se obtiene cuando se suma un termino menor a la tolerancia (TOL).

Implementar un algoritmo que aproxime $f(x) = e^x$ dados los valores de x y TOL . El algoritmo debe informar el grado del polinomio utilizado (valor de N).

Ejercicio 22

La raiz cuadrada de un numero positivo A puede aproximarse mediante un proceso iterativo que genera terminos según la siguiente formula:

$$R_i = \begin{cases} 1 & i=1 \\ \frac{R_{i-1} + \frac{A}{R_{i-1}}}{2} & i>1 \end{cases}, \text{ donde } i \text{ es el numero de la iteracion.}$$

El proceso concluye cuando la diferencia entre dos terminos sucesivos es menor que una cota de fijada de antemano (*TOL*). Desarrollar un algoritmo que aproxime a la raiz de un numero *A*, dada la cota. El algoritmo debe mostrar por pantalla, el valor real, el valor aproximado y la cantidad de iteraciones.

Ejercicio 23

La serie de Fibonacci, es la serie en la cual cada termino se define iterativamente como la suma de los dos terminos anteriores, los primeros dos terminos son iguales a 1.

$$F_i = \begin{cases} 0 & i=0 \\ 1 & i=1 \\ F_{i-2} + F_{i-1} & i>1 \end{cases}$$

Desarrollar un programa que reciba un numero natural e informe si pertenece o no a la serie de Fibonacci.

Ejercicio 24

Dados dos números naturales (incluido el cero), obtener su producto por sumas sucesivas.