

UT1.A00

Introducción a la programación

Actividades de programación	2
Parte 1	2
Ejercicio 1.1	2
Ejercicio 1.2	2
Ejercicio 1.3	2
Parte 2.	3
Ejercicio 2.1	3
Ejercicio 2.2	3
Ejercicio 2.3	3
Ejercicio 2.4	3
Ejercicio 2.5	3
Ejercicio 2.6	3
Ejercicio 2.7	3
Ejercicio 2.8	3
Ejercicio 2.9	4
Ejercicio 2.10	4
Ejercicio 2.11	4
Ejercicio 2.12	4
Ejercicio 2.13	4
Ejercicio 2.14	4
Ejercicio 2.15	4
Ejercicio 2.16	5
Ejercicio 2.17	5
Ejercicio 2.18	5

Actividades de programación

Parte 1

Ejercicio 1.1

El objetivo del ejercicio es crear un sistema de calificaciones, como sigue:

El usuario proporcionará un valor entre 0 y 10.

Si está entre 9 y 10: imprimir una A

Si está entre 8 y menor a 9: imprimir una B

Si está entre 7 y menor a 8: imprimir una C

Si está entre 6 y menor a 7: imprimir una D

Si está entre 0 y menor a 6: imprimir una F

cualquier otro valor debe imprimir: Valor desconocido

Por ejemplo:

Proporciona un valor entre 0 y 10:

A

Puedes utilizar el IDE de tu preferencia para codificar la solución y después pegar tu solución en esta herramienta.

Ejercicio 1.2

Solicitar al usuario dos valores:

- numero1 (int)
- numero2 (int)

Se debe imprimir el mayor de los dos números (la salida debe ser idéntica a la que sigue):

Proporciona el numero1:

Proporciona el numero2:

El numero mayor es:<numeroMayor>

¿Cuál es el código del requerimiento solicitado?

Ejercicio 1.3

Iterar un rango de 0 a 10 e imprimir sólo los números divisibles entre 3

Parte 2.

Ejercicio 2.1

Definir una función `max()` que tome como argumento dos números y devuelva el mayor de ellos. (Es cierto que python tiene una función `max()` incorporada, pero hacerla nosotros mismos es un muy buen ejercicio.

Ejercicio 2.2

Definir una función `max_de_tres()`, que tome tres números como argumentos y devuelva el mayor de ellos.

Ejercicio 2.3

Definir una función que calcule la longitud de una lista o una cadena dada. (Es cierto que python tiene la función `len()` incorporada, pero escribirla por nosotros mismos resulta un muy buen ejercicio.

Ejercicio 2.4

Escribir una función que tome un carácter y devuelva `True` si es una vocal, de lo contrario devuelve `False`.

Ejercicio 2.5

Escribir una función `sum()` y una función `multip()` que sumen y multipliquen respectivamente todos los números de una lista. Por ejemplo: `sum([1,2,3,4])` debería devolver 10 y `multip([1,2,3,4])` debería devolver 24.

Ejercicio 2.6

Definir una función `inversa()` que calcule la inversión de una cadena. Por ejemplo la cadena "estoy probando" debería devolver la cadena "odnaborp yotse"

Ejercicio 2.7

Definir una función `superposicion()` que tome dos [listas](#) y devuelva `True` si tienen al menos 1 miembro en común o devuelva `False` de lo contrario. Escribir la función usando el bucle `for` anidado.

Ejercicio 2.8

Definir una función `generar_n_caracteres()` que tome un entero `n` y devuelva el caracter multiplicado por `n`. Por ejemplo: `generar_n_caracteres(5, "x")` debería devolver "xxxxx".

Ejercicio 2.9

Definir un histograma procedimiento() que tome una lista de números enteros e imprima un histograma en la pantalla. Ejemplo: procedimiento([4, 9, 7]) debería imprimir lo siguiente:

Ejercicio 2.10

La función max() del ejercicio 1 (primera parte) y la función max_de_tres() del ejercicio 2 (primera parte), solo van a funcionar para 2 o 3 números. Supongamos que tenemos más de 3 números o no sabemos cuántos números son. Escribir una función max_in_list() que tome una lista de números y devuelva el más grande.

Ejercicio 2.11

Escribir una función mas_larga() que tome una lista de palabras y devuelva la más larga.

Ejercicio 2.12

Escribir una función filtrar_palabras() que tome una lista de palabras y un entero n, y devuelva las palabras que tengan más de n caracteres.

Ejercicio 2.13

Escribir un programa que le diga al usuario que ingrese una cadena. El programa tiene que evaluar la cadena y decir cuantas letras mayúsculas tiene.

Ejercicio 2.14

Construir un pequeño programa que convierta números binarios en enteros.

Ejercicio 6

Escribir un pequeño programa donde:

- Se ingresa el año en curso.
 - Se ingresa el nombre y el año de nacimiento de tres personas.
 - Se calcula cuántos años cumplirán durante el año en curso.
 - Se imprime en pantalla.
-

Ejercicio 2.15

Definir una tupla con 10 edades de personas.

Imprimir la cantidad de personas con edades superiores a 20.

Puedes variar el ejercicio para que sea el usuario quien ingrese las edades.

Ejercicio 2.16

Definir una lista con un conjunto de nombres, imprimir la cantidad de comienzan con la letra a.

También se puede hacer elegir al usuario la letra a buscar. (Un poco mas emocionante)

Ejercicio 2.17

Crear una función contar_vocales(), que reciba una palabra y cuente cuantas letras "a" tiene, cuantas letras "e" tiene y así hasta completar todas las vocales.

Se puede hacer que el usuario sea quien elija la palabra.

Ejercicio 2.18

Escriba una función es_bisiesto() que determine si un año determinado es un año bisiesto. Un año bisiesto es divisible por 4, pero no por 100. También es divisible por 400