

Ejercicios Tema 7. Flujos de Entrada y Salida

EJERCICIO 1.

Crea un programa que lea 4 datos introducidos por teclado y los almacene dentro de una stack. A continuación, el mismo programa debe volver a imprimir estos mismos cuatro datos pero con el siguiente formato:

- El primer dato, que es un valor entero de 8 dígitos, debe imprimirse con una cantidad mínima de caracteres de 10, ajustado a la derecha.
- El segundo y tercer dato son valores flotantes. El segundo dato debe imprimir un total de 6 dígitos, mientras que el tercero debe imprimir como máximo 3 dígitos en notación científica. Ambos datos deben imprimirse con justificación a la izquierda.
- Por último, el cuarto dato se corresponde con un string. Este dato debe volver a imprimirse pero utilizando comillas.

EJERCICIO 2.

Crea un programa que lea 5 datos introducidos por teclado y los almacene en una cola (queue). Posteriormente, imprime estos datos aplicando el siguiente formato:

- El **primer dato**, una cadena de texto con varios espacios ajustado a la izquierda en un espacio de 20 caracteres.
- El **segundo dato**, un número entero que debe dividirse entre 10 y el resultado debe estar justificado a la derecha con un ancho de 10 caracteres y precisión de 1 decimal.
- El **tercer y cuarto dato**, números flotantes, deben imprimirse con 2 y 5 cifras decimales respectivamente, ambos alineados a la izquierda.
- El **quinto dato**, una palabra clave, debe imprimirse entre corchetes.

EJERCICIO 3.

Crear una clase Vector que tenga dos atributos públicos: x: float e y: float y dos funciones amigas de operadores de flujo de entrada y salida. Crear un programa principal que, a través del uso de los operadores, ingrese datos introducidos por teclado y los vuelva a imprimir utilizando el siguiente formato:

(x, y)

Teniendo como mínimo 3 decimales en cada valor, ajustando ambos valores a la derecha, con una cantidad mínima de caracteres de 10 por atributo.

EJERCICIO 4.

Utilizando la clase Vector, crear un programa que obtenga por teclado una serie de objetos de esta clase y los almacene dentro de una fila. A continuación, cree un objeto que cree un fichero de salida llamado "salida.txt" que almacene cada uno de los objetos obtenidos por teclado.

El fichero generado debe tener la siguiente forma:

1. $x = 1.0e+1$ $y = 2.5e+1$
2. $x = 2.3e+1$ $y = 0.2e+1$
3. etc ...

EJERCICIO 5.

Modifica la clase Vector para que tenga tres atributos públicos: x: float, y: float, z: float.

Implementa operadores de entrada y salida como funciones amigas, de manera que se puedan introducir e imprimir objetos con el siguiente formato:

$x - y - z$

Cada valor deberá:

- Tener exactamente 4 cifras decimales, incluso si son ceros.
- Estar justificado a la derecha dentro de un campo de 12 caracteres.

Luego, crea un programa principal que solicite al usuario introducir al menos 2 objetos Vector y los cree un fichero cuyo contenido corresponde a cada vector introducido con el formato mencionado.

EJERCICIO 6. Usando la clase del ejercicio anterior (Vector), escribe un programa que:

- Lea una lista de vectores desde un fichero llamado entrada.txt, donde cada línea tenga el formato:
- Calcule el módulo de cada vector y los almacene en otra lista.
- Escriba un fichero resultado.txt con el siguiente formato:

1. $x = 1.0000e+1$ $y = 2.0000e+1$ $z = 3.0000e+1$ | módulo = $3.7417e+1$

2. $x = \dots$ $y = \dots$ $z = \dots$ | módulo = ...

Todos los números deben imprimirse en notación científica con exactamente 4 cifras decimales.