

## Examen 1ª Evaluación DAMIoT

### Nombre y apellidos:

**Nota:** Lee atentamente el enunciado. Cualquier duda sobre el enunciado pregunta a la profesora. Cada pregunta lleva un ejemplo de ejecución del programa.

En el aula virtual tienes un esqueleto de cada ejercicio. Cuando termines, crea un fichero .zip o .rar con los 3 ejercicios y súbelos al aula virtual.

### EJERCICIO 1 (2,5 puntos) - Programa en C o PSeInt

Escribe un programa (puedes hacerlo en C o PSeInt) que solicite números al usuario todo el tiempo. Cada número introducido debe ser mayor que el anterior. En el momento que se introduzca un número menor o igual que el anterior se dejará de solicitar números y se imprimirá el siguiente resumen.

- Total de números introducidos, sin contar el último número que nos hizo salir del bucle.
- Total de números mayores que cero, sin contar el último número que nos hizo salir del bucle.
- Total de números pares, sin contar el último número que nos hizo salir del bucle.
- La suma de todos los valores pares, sin contar el último número que nos hizo salir del bucle.

No puedes utilizar arrays para resolver el problema.

No sabes la cantidad de números que va a introducir el usuario.

Puedes utilizar funciones si quieres, da igual. No lo tengo en cuenta.

Observa que el programa como mínimo pedirá dos números al usuario.

**Alternativa:** Si no sabes cómo pedirle números al usuario hasta que el valor introducido sea menor que el anterior, haz el ejercicio solicitando únicamente 5 números. Obtendrás menos puntuación. -0,5 puntos.

#### Ejemplo de ejecución 1:

Dime un número: -10

Dime un número: -1

Dime un número: 3

Dime un número: 4

Dime un número: 5

Dime un número: 0

RESUMEN

Total de números introducidos: 5

Total de números mayores que cero 3

Total de números pares: 2

Suma de valores pares: -6

### Ejemplo de ejecución 2:

Dime un número: 5

Dime un número: 2

RESUMEN

Total de números introducidos: 1

Total de números mayores que cero 1

Total de números pares: 0

Suma de valores pares: 0

## EJERCICIO 2 (2,75 puntos) - Programa en C

Dada la siguiente estructura:

```
struct producto{  
    char codigo[4]; //3 caracteres + \0  
    float precio;  
}
```

Vas a guardar información de **3 productos en un array de estructura producto**. Para ello mostrarás al usuario el siguiente menú de opciones. El programa está continuamente ejecutándose hasta que el usuario introduce la opción de salir. Si introduce cualquier opción inválida se le informa al usuario.

Crea un menú con las siguientes opciones:

1. **Insertar productos:** Se insertarán los 3 productos en esta opción. Se solicitarán el código y precio del producto. El precio debe ser un número >0. El código debe empezar por la letra 'E' mayúscula. Supondremos que el usuario no introduce códigos repetidos, no hace falta que lo compruebes. En esta opción es obligatorio que el usuario inserte los 3 productos correctamente, no pararemos de solicitar datos hasta que estén los 3 productos. Fíjate que sólo podrás ejecutar esta opción una vez, las siguientes veces informa al usuario de que están todos los productos.
2. **Imprimir:** Imprime los productos (si los hay). Informar si no hay productos.
3. **Consultar precio de un producto:** Solicita el código de un producto e imprime por pantalla su precio. Informa al usuario si no hay productos o no existe el código.
4. **Salir.**

Puedes utilizar las funciones de la librería string.h.

Puedes utilizar funciones si quieres, da igual. No lo tengo en cuenta.

Cuando introduzcas códigos de producto no pongas más de 3 caracteres porque es el máximo que puede guardar el array. Serían válidos por ejemplo E10, E11, E12. Inválidos 123, ABC, A12.

### Ejemplo de ejecución

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 5

La opción no existe

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 2

No hay productos

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 3

No hay productos

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 1

Introduce un código del producto 1

> T10

Introduce el precio del producto 1

> 10.5

Datos erróneos

Introduce un código del producto 1

> E10

Introduce el precio del producto 1

> 10.5

Introduce un código del producto 2

> E11

Introduce el precio del producto 2

> 12

Introduce un código del producto 3

> E13

Introduce el precio del producto 3

> -10

Datos erróneos

Introduce un código del producto 3

> E13

Introduce el precio del producto 3

> 10

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 2

Código: E10, Precio: 10.500000

Código: E11, Precio: 12.000000

Código: E13, Precio: 10.000000

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 3

Introduce un código

> E11

Código: E11, Precio: 12.000000

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 3

Introduce un código

> E17

El código de producto no existe

Introduce una opción:

1.Insertar

2.Imprimir

3.Consultar precio

4.Salir

> 1

No se pueden insertar más productos

Introduce una opción:

- 1.Insertar
  - 2.Imprimir
  - 3.Consultar precio
  - 4.Salir
- > 4  
FIN

## EJERCICIO 3 (2,75 puntos) - Programa en C

Crea un array bidimensional 3 x 3 de valores enteros con valores aleatorios. El programa no solicita nada al usuario. Deberá mostrar la siguiente información por pantalla.

- Imprimirá los valores del array.
- Imprimir la media del array. Recuerda que para calcular la media haz la suma de todos los valores y lo divides entre el tamaño del array.
- Imprimir el valor mínimo de cada fila.

Para ello tendrás que utilizar las siguientes funciones:

- Función **inicializaArray**: No devuelve nada. Recibe como parámetro un array 3x3. Inicializa cada posición del array con valores aleatorios del 0 al 5.
- Función **imprimeArray**: Imprime el array por pantalla con este formato:

```
1 2 3
6 1 0
2 4 1
```

- Función **calculaMedia**: Recibe como parámetro el array y la variable 'media' pasada por referencia. Esta función calcula la media de todos los valores del array. Ten en cuenta que podría tener decimales. Después de llamar a esta función tendrás que imprimir el valor de media en el main.
- Función **minimoFila**: Devuelve una estructura como la que verás a continuación. Recibe como parámetro un array. Devuelve en la estructura el valor mínimo de cada fila. Después de llamar a esta función, en el main, deberás recoger la estructura que devuelve esta función e imprimirla por pantalla.

```
struct minimo{
    int fila0;
    int fila1;
    int fila2;
};
```

No se podrán utilizar variables globales.

Tendrás que añadir las librerías para calcular los valores aleatorios y la semilla.

### Ejemplo de ejecución 1:

IMPRIMO ARRAY



C/ Carlos III, 3  
30201 CARTAGENA  
TEL: 968 321 301  
30019702@murciaeduca.es  
www.cifpcarlos3.es



3 5 5

4 4 2

1 1 1

Media: 2.888889

Fila 0:3 - Fila 1:2 - Fila 2:1

## Ejemplo de ejecución 2:

IMPRIMO ARRAY

5 0 5

1 2 4

0 0 3

Media: 2.222222

Fila 0:0 - Fila 1:1 - Fila 2:0