

PROGRAMACIÓN I - GRUPO A

PRÁCTICA Nº4

17 de noviembre 2022

INSTRUCCIONES

1. La entrega de los ejercicios de la práctica sólo se admitirán a través de la **actividad** disponible en el campus virtual de la asignatura de Programación I antes de la hora de finalización de la sesión de prácticas.
2. Aunque las prácticas se pueden realizar en grupos de dos integrantes, para su evaluación, **ambos deberán hacer la entrega a través de su campus virtual**. En otro caso, la práctica quedará sin evaluar y supondrá un 0 en su calificación.
3. Se deben entregar tantos ficheros como ejercicios y/o apartados tenga el enunciado de la práctica. En este caso concreto se debe entregar **un único fichero en formato .txt o .cpp, sin comprimir**.
4. Cada fichero entregado debe **incluir el nombre de los integrantes del equipo**.
5. Durante el desarrollo de la sesión práctica se debe seguir el procedimiento de trabajo en la sala asignada para realizar las actividades evaluables. Recordad que es **imprescindible GRABAR** toda la sesión y, además, **COMPARTIR el escritorio y ACTIVAR la cámara web y el micrófono**.
6. También se debe acceder a la **SALA DE PRÁCTICA**, desde donde se darán indicaciones y se resolverán dudas generales. Si lo consideráis necesario, utilizad **el chat** de esta sala para **comunicaros con el profesor/a**.
7. **El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su descalificación**.

EJERCICIO

Escriba un programa en C++11, **PracticaA4**, utilizando los **tipos de datos**, las **estructuras de control** y las **funciones sin parámetros o con paso por valor** necesarias, que muestre por pantalla un **menú** con las siguientes opciones:

1. **Titulos de cuentos infantiles.**
2. **Operar con una matriz.**
3. **Determinar si la palabra/frase es palindroma.**
4. **Salir programa.**

El menú deberá estar en una **función que no tendrá parámetros**, pero que debe tener un **valor de retorno**, la **opción seleccionada por el usuario**. Si la **opción indicada por el usuario no es válida**, se mostrará el mensaje **“La opción no es válida. Por favor, vuelva a seleccionar otra opción”** y **presentará de nuevo el menú anterior** para que el usuario pueda **repetir el proceso** seleccionando otra opción **(1 punto)**.

Siempre que se seleccione cualquiera de las **tres primeras opciones** el programa, desde el **main**, deberá realizar la **llamada a una función sin parámetros o con paso por valor** para ejecutar dicha operación y, después, volverá a **presentar de nuevo el menú anterior** para que el usuario pueda **repetir el proceso** seleccionando otra opción. Si el usuario selecciona la **opción 4**, el programa terminará mostrando el mensaje: **“Gracias y hasta pronto” (0,5 puntos)**.

Importante: Las diferentes opciones deberán ser resueltas de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general y no particular** para unos **determinados datos**.

La descripción de las opciones del menú es la siguiente:

1. Titulos de cuentos infantiles. (3 puntos)

Esta opción será la **función sin parámetros datosCuentos**, que consistirá en crear un **array de dimensión 5** para guardar **datos de cuentos infantiles** que se obtendrán de los **títulos** que se solicitarán al usuario **(1 punto)**.

Cada elemento del array será de **tipo struct** y tendrá que almacenar **(0,5 puntos)**:

- el **título** del cuento.
- el **número de palabras** del título.
- El **número de caracteres** del título (incluidos los espacios en blanco).

El cálculo del **número de palabras** se tendrá que hacer con una **función**, **totPalabra**, que **recibirá como parámetro el título de un cuento (tipo *string*)** y **devolverá el número de palabras** calculado **(1 punto)**.

Finalmente, se mostrará por pantalla el contenido del **array** utilizando un **bucle for(:)** **(0,5 puntos)**.

Por ejemplo, supongamos que uno de los títulos tecleadas por el usuario y almacenado en el array es: **“El gato con botas”**, la ejecución de esta opción mostrará en pantalla:

El cuento titulado: El gato con botas tiene 4 palabras y su longitud es 17

De la misma manera se procederá con el resto de títulos almacenados en el array.

```
INTRODUZCA LOS DATOS DE 5 CUENTOS
Introduzca el titulo numero 1 EL gato con botas
Introduzca el titulo numero 2 Blacanieves
Introduzca el titulo numero 3 Los tres cerditos
Introduzca el titulo numero 4 Los aristogatos
Introduzca el titulo numero 5 Pulgarcito

LOS DATOS DE LOS CUENTOS SON:
El cuento titulado: El gato con botas tiene 4 palabras y su longitud es 17
El cuento titulado: Blacanieves tiene 1 palabras y su longitud es 11
El cuento titulado: Los tres cerditos tiene 3 palabras y su longitud es 17
El cuento titulado: Los aristogatos tiene 2 palabras y su longitud es 15
El cuento titulado: Pulgarcito tiene 1 palabras y su longitud es 10
```

2. Operar con una matriz. (4 puntos)

Esta opción será la **función sin parámetros operarMatriz**, que consistirá en solicitar al usuario **números enteros positivos**, comprendidos **entre 0 y 9**, para almacenarlos en una **matriz de dimensión 4x4, matriz4x4**. Habrá que validar que los valores dados por el usuario cumplen con las especificaciones del enunciado, en otro caso, se mostrará un mensaje informativo al usuario y se volverá a solicitar otro número para dicha posición de la matriz **(1 punto)**.

Una vez creada la matriz anterior, **matriz4x4** con los valores correctos (valores comprendidos entre 0 y 9), se realizarán las siguientes operaciones.

- Mostrar el contenido de la matriz por pantalla en formato de **filas y columnas** delimitada por el **carácter ‘|’** utilizando un bucle **for(:)** **(0,75 puntos)**.

- Generar una **nueva matriz, matrizNew**, de dimensión 3 para almacenar **tres elementos** de **tipo struct** con tres campos:
 - **suma**: un **valor entero** para almacenar una suma;
 - **espar**: un **valor bool** para indicar si es par (true) o impar (false);
 - **zona**: un **valor de tipo string** para guardar la palabra que indica la zona de la matriz (superior, diagonal, inferior).

El contenido de cada uno de los elementos de la nueva matriz se obtendrá de la siguiente forma:

- Primer elemento: el campo **suma** contendrá la **suma de los números situados en la parte superior de la diagonal principal**. El campo **espar** contendrá el valor **true**, si el valor de **suma** es **PAR**, o valor **false** si es **IMPAR**. Por último, en el campo **zona** se guardará la palabra **“superior”** (0,5 puntos).
- Segundo elemento: el campo **suma** contendrá la **suma de los números en la diagonal principal**. El campo **espar** contendrá el valor **true**, si el valor de **suma** es **PAR**, o valor **false** si es **IMPAR**. Por último, en el campo **zona** se guardará la palabra **“diagonal”** (0,5 puntos).
- Tercer elemento: el campo **suma** contendrá la **suma de los números situados en la parte inferior de la diagonal principal**; El campo **espar** contendrá el valor **true**, si el valor de **suma** es **PAR**, o valor **false** si es **IMPAR**. Por último, en el campo **zona** se guardará la palabra **“inferior”** (0,5 puntos).
- Mostrar por pantalla el contenido del **array** creado en el punto anterior utilizando un bucle **for(:)**. (0,75 punto)

Por ejemplo, supongamos que los números que introduce el usuario son:

1, 2, 3, 4, 5, 6, **77**, 7, 8, 9, 0, 1, 2, 3, 4, 5, 6

Donde el **77** es un valor NO VÁLIDO. La ejecución del programa mostrará en pantalla:

```
INTRODUZCA LOS DATOS DE LA MATRIZ
unaMatriz [0, 0] = 1
unaMatriz [0, 1] = 2
unaMatriz [0, 2] = 3
unaMatriz [0, 3] = 4
unaMatriz [1, 0] = 5
unaMatriz [1, 1] = 6
unaMatriz [1, 2] = 77
unaMatriz [1, 3] = 7
unaMatriz [2, 0] = 8
unaMatriz [2, 1] = 9
unaMatriz [2, 2] = 0
unaMatriz [2, 3] = 1
unaMatriz [3, 0] = 2
unaMatriz [3, 1] = 3
unaMatriz [3, 2] = 4
unaMatriz [3, 3] = 5
unaMatriz [3, 3] = 6
```

```
LA MATRIZ DATO ES
| 1 2 3 4 |
| 5 6 7 8 |
| 9 0 1 2 |
| 3 4 5 6 |

LA MATRIZ GENERADA ES
| 26 1 superior|
| 14 1 diagonal|
| 26 1 inferior|
```

3. Determinar si la palabra/frase es palíndroma. (1,5 puntos)

Esta opción será la función **esPalindroma**, que recibirá **una palabra o frase** (tipo **string**) y devolverá un valor de tipo **bool** para indicar si es palíndroma o no (**1 punto**).

La palabra o frase se deberá solicitar en el **main** y la función devolverá un valor lógico (**bool**) para que en el **main** se muestre un mensaje indicando si la palabra o frase era palíndroma o no (**0,5 puntos**).

Se dice que una palabra o frase es palíndroma si se lee igual en un sentido que en otro (por ejemplo; ANA, AMOR A ROMA).

NOTA IMPORTANTE: No se tienen que considerar los espacios en blanco, es mejor eliminarlos antes de procesar la palabra/frase (se puede utilizar el método **erase**). Además, como se diferencia entre mayúsculas y minúsculas, se puede escribir siempre en mayúsculas o en minúsculas toda la palabra/frase.

```
1. Titulos de cuentos infantiles.
2. Operar con una matriz.
3. Determinar si la palabra/frase es palindroma.
4. Salir programa.

Teclee la opcion: 3

Por favor, teclee un texto: amor a roma

La palabra/frase es PALINDROMA.
```

```
1. Titulos de cuentos infantiles.
2. Operar con una matriz.
3. Determinar si la palabra/frase es palindroma.
4. Salir programa.

Teclee la opcion: 3

Por favor, teclee un texto: Hoy es jueves

La palabra/frase NO es PALINDROMA.
```