

PROGRAMACIÓN I - GRUPO C

PRÁCTICA Nº2

24 de octubre 2023

INSTRUCCIONES

1. Se debe acceder a la **SALA DE GRUPO** del campus virtual de vuestro equipo de trabajo.
2. 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 COMPARTIR la pantalla completa**.
3. La entrega de la práctica sólo se admitirá a través de la **actividad** disponible en el campus virtual de la asignatura de **Prácticas de Programación I antes de la hora de finalización de la sesión de prácticas**.
4. Se debe entregar **un único fichero** en formato **.txt o .cpp**, **sin comprimir**, con el nombre de **P2GC**.
5. El fichero entregado debe **incluir el nombre de los integrantes del equipo**.
6. Aunque las prácticas se realizan 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**.
7. **El incumplimiento de alguna de las instrucciones sobre la realización/entrega de la Práctica supondrá su descalificación**.

A. TEST [30% puntuación] – 20 min

Realizar de **forma individual** desde el campus virtual de la asignatura accediendo al apartado **Enunciado PII/Test Práctica N2-PI Grupo C (30%)**.

- **No se permite ningún tipo de material de apoyo.**
- El test consta de **10 preguntas** que se mostrarán de **una en una en orden aleatorio**.
- Puntuación por pregunta:
 - Correcta +1 punto
 - No contestado 0 puntos
 - Incorrecta -0.25 puntos.
- **Tiempo 20 minutos máximo.**
- Una vez iniciado el test, se debe terminar de una sola vez.
- Si hay algún problema técnico, comunicarlo al profesor/a.

B. EJERCICIO PRÁCTICO [70% puntuación] – 150 min

Escriba un programa en C++11, **P2GC**, utilizando los tipos de datos, las operaciones de entrada y salida y las estructuras de control vistas hasta ahora, que muestre por pantalla un menú con las siguientes opciones:

1. **Validar fecha.**
2. **Determinar las cifras pares/impares de un número.**
3. **Determinar letras y palabras de un texto.**
4. **Salir programa.**

Siempre que se seleccione cualquiera de las primeras opciones el programa deberá realizar dicha operación y volver a presentar de nuevo el menú anterior para que el usuario pueda repetir el proceso seleccionando otra opción. Si el usuario selecciona la **última opción**, el programa terminará mostrando el mensaje: **“Gracias y hasta pronto”**.

Si la opción indicada por el usuario no es ninguna de las anteriores, se mostrará el mensaje **“La opción no es válida. Por favor, vuelva a seleccionar otra opción”**.

(1,5 puntos)

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

La descripción de las opciones del menú es la siguiente:

1. Validar fecha (3 puntos)

Esta opción consistirá en **solicitar una fecha al usuario, solicitando el DIA, MES y AÑO** (puede almacenarse en un variable de tipo **struct**), para **determinar si se trata de una fecha válida o no**. Finalmente, se **mostrará la fecha introducida por pantalla en formato DIA/MES/AÑO** indicando al usuario si es **CORRECTA o INCORRECTA**.

Por ejemplo, si la fecha tecleada por el usuario es 24 10 2023, se mostrará por pantalla:

24/10/2023 en una fecha CORRECTA

Mientras que, si la fecha tecleada por el usuario es 29 2 2023, se mostrará por pantalla:

28/2/2023 en una fecha INCORRECTA

Información adicional: La siguiente tabla señala los días que tiene cada mes del año:

Número	Mes	Duración
1	Enero	31 días
2	Febrero	28 días o 29 en los años bisiestos
3	Marzo	31 días
4	Abril	30 días
5	Mayo	31 días
6	Junio	30 días
7	Julio	31 días
8	Agosto	31 días
9	Septiembre	30 días
10	Octubre	31 días
11	Noviembre	30 días
12	Diciembre	31 días

Un año es **bisiesto** si es **divisible por 4**, **excepto cuando es divisible por 100, a no ser que sea divisible por 400**. Por ejemplo: el año 1900 no fue bisiesto ya que, aunque es divisible por 4 también lo es por 100. Sin embargo, el año 2000 si fue bisiesto, por ser divisible por 400.

2. Determinar las cifras pares/impares de un número (2,5 puntos)

Esta opción consistirá en solicitar un **número entero positivo al usuario** para determinar sus cifras y guardar en una variable de tipo **struct** los campos que seguidamente se describen y cuyo contenido se debe de mostrar por pantalla.

La variable de tipo **struct** debe contener los siguientes campos:

- **cifrasPares**: campo que almacenará el número de **cifras pares** que tiene el número solicitado.
- **cifrasImpares**: campo que almacenará el número de **cifras impares** que tiene el número solicitado.
- **totalCifras**: campo que almacenará el **total de cifras** que tiene el número solicitado.

El proceso debe realizar las siguientes tareas:

- Solicitar un **único número entero positivo**.
- Determinar, a través de las operaciones de **división (/)** y **resto de división entera (%)**, las cifras que forman el número y procesar cada una de ellas para obtener la información de la **struct** descrita anteriormente:
 - Determinando si se debe actualizar alguno de los campos **cifrasPares** o **cifrasImpares**, según corresponda.
 - Incrementando el campo **totalCifras**.
- Mostrar por pantalla el número solicitado al usuario y la información obtenida sobre sus cifras (contenido de los campos **cifrasPares**, **cifrasImpares** y **totalCifras** de la **struct**).

3. Determinar letras y palabras de un texto (2 puntos)

Esta opción consistirá en solicitar al usuario un texto (**string**) para determinar el **número de vocales (sin diferenciar entre mayúsculas y minúsculas), el número de palabras y el número de caracteres distintos del espacio blanco contiene dicho texto**. El resultado del análisis realizado servirá para incrementar el total de vocales y de palabras que se incluirá en una variable de tipo **struct** y cuyo contenido se deberá de mostrar por pantalla al final de proceso.

La variable de tipo **struct** debe contener los siguientes campos:

- **totalVocales**: campo que almacenará el **total de vocales** incluidas en el texto tecleado.
- **totalPalabras**: campo que almacenará el **total de palabras** incluidas en el texto tecleado.
- **totalCaracteres**: campo que almacenará el **total de caracteres** (sin contar los espacios en blanco) tiene en el texto tecleado.
-

El proceso debe realizar las siguientes tareas:

- Solicitar al usuario un **texto (puede incluir espacios en blanco)**.
- Analizar el texto para incrementar el correspondiente campo de la variable **struct** donde se lleva la cuenta para el número de vocales y de palabras existentes en el texto.
- Finalmente, se mostrará por pantalla el contenido de la variable **struct** (contenido de los campos que son los contadores de vocales y palabras del texto).

Nota: las funciones **toupper/tolower** permiten convertir un carácter (tipo **char**) a mayúscula/minúscula, respectivamente.

RUBRICA DE CALIFICACIÓN

- El 30% de la nota de la práctica 2, simulacro de examen, vendrá dado por el test (10 preguntas, 20 minutos). Esta puntuación será individual.
- El 70% restante será de la práctica entregada (misma nota para cada integrante de manera general), siguiendo la siguiente rúbrica para cada apartado:

Se debate en equipo sobre la estrategia a seguir, haciendo un boceto de la estructura que tendrá el programa (a papel, en la siguiente hoja).	1 punto
El programa muestra las diferentes opciones del menú , permitiendo al usuario seleccionar una opción, y repitiéndose hasta que se pulse salir.	1.5 puntos
Opción 1: Validar fecha	
Petición y validación de los valores independientes para el día, mes y año	0,75 puntos
Comprobación de validez entre día/mes considerando el año bisiesto cuando sea adecuado.	1,5 puntos
Visualización del resultado.	0,75 puntos
Opción 2: Determinar las cifras pares/impares de un número	
Determinación de las cifras del número	1 punto
Definir struct . Procesar la cifra y contabilizarla en el campo correspondiente de la struct	1 punto
Visualización del número y del contenido de la struct	0.5 puntos
Opción 3: Determinar letras y palabras de un texto	
Solicitar texto de tipo string	0.5 puntos
Definir struct . Contabilizar vocal, palabra o carácter en la struct	1 punto
Visualización el contenido de la struct	0.5 puntos

Además, cada apartado de la parte práctica, se evaluará considerando lo siguiente:

	% máx. (*)
El elemento evaluable no compila o no se asemeja a lo que se pide	0%
El elemento evaluable no se aproxima suficientemente a lo pedido	40%
El elemento evaluable se aproxima suficientemente a lo pedido	60%
El elemento evaluable funciona correctamente y las estrategias y elementos de código elegidos son adecuados.	100%

(*) El porcentaje (% max.) representa el valor máximo sobre el que se evalúa el elemento indicado.