

ALUMNO:

EPS -Ingeniería Informática

Asignatura: G0460006 Programación I

Curso: 2023/2024
Semestre: 1º

Examen: Final
Convocatoria: Ordinaria

Fecha: 15-1-2024 (9:00-11:30)

Parte Práctica (10 puntos; 70% nota final) - Tiempo: 2h10m

Los ficheros correspondientes a los ejercicios se deben entregar en la actividad correspondiente a través del campus antes de la finalización del tiempo establecido. Cada fichero se llamará EjercicioX, donde la X será el número de dicho ejercicio. Los únicos formatos válidos serán **txt** o **cpp**, siempre **sin comprimir**.

Criterios generales de evaluación

| | % máx. (*) |
|--|------------|
| Funciones/Métodos: Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple. | 40% |
| Tipos de datos y variables: | |
| • Uso de variables globales (fuera del ámbito de una función). | 0% |
| • Si no se usan los tipos contenedor vistos en clase (std::array; std::vector; std::set; std::string, etc.) para las variables que lo necesiten. | 0% |
| • Si no se usan punteros inteligentes (std::unique_ptr; std::shared_ptr) cuando sea necesario. | 0% |
| • Si se utilizan punteros clásicos. | 0% |
| No utilizar funciones/métodos cuando sea apropiado o se indique de forma explícita en el apartado/ejercicio de la actividad/examen . | 40% |
| El programa no compila o no se asemeja a lo pedido. | 0% |
| Si no se cumplen los criterios de entrega indicados en la actividad/examen . | 0% |
| Si los apartados/ejercicios de la actividad/examen son resueltos de forma NO algorítmica , es decir, proponiendo una solución particular para unos determinados datos/valores en lugar de ser una solución general independiente de los datos/valores. | 0% |

Criterios particulares de evaluación

| | % 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 (% máx.) representa el valor máximo sobre el que se evalúa el elemento indicado.

IMPORTANTE:

- Todos los ejercicios del examen deberán ser resueltos de forma **algorítmica**, es decir, la **solución** propuesta tendrá que ser **general** y **no particular** para unos determinados datos/valores.
- Se recomienda una primera lectura del examen completo para planificar la realización del examen. Y una segunda lectura detallada antes de la realización de cada uno de los ejercicios propuestos.

Ejercicio 1 (5 puntos)

Escriba un programa en C++11, *Ejercicio1*, utilizando los tipos de datos, las **estructuras de control** y las **funciones** necesarias para llevar el control de material de un almacén. **El almacén en cuestión estará formado por un número indeterminado de cajas, pudiendo almacenar como máximo 5 elementos por caja** (En cada caja únicamente podrá haber elementos iguales).

- Implementar el mecanismo que crea oportuno para representar en la memoria del computador un elemento del inventario. Dicho *Elemento* deberá ser capaz de almacenar una descripción textual y una referencia numérica entera. **[0,25 puntos]**
- Declarar una variable de etiqueta *Almacen* que pueda albergar la información descrita al comienzo del enunciado. **[0,5 puntos]**
- Desarrollar en la función principal el código necesario para permitir el registro del material introducido mediante teclado por un usuario en el *Almacén*, atendiendo las siguientes indicaciones:
 - El programa solicitará de forma cíclica al usuario la información necesaria para describir el *Elemento*, el valor numérico entero para referenciarlo y el número de elementos a almacenar **[1 punto]**
 - El **proceso de captura de información finalizará** cuando ocurra alguna de las siguientes circunstancias: **[0,75 puntos]**
 - La **descripción** introducida por el usuario **contenga en alguna parte el texto “Fin captura”**. Por ejemplo:
 - La descripción “Lapiceros de colores” no debería hacer finalizar el proceso de captura
 - La descripción “Lapiceros Fin captura de colores” si debería hacer finalizar el proceso de captura.
 - Cuando la **referencia** numérica del Elemento sea **menor o igual a cero**.
 - Cuando el **número de elementos** introducidos sea **menor o igual a cero**.



- El código debe contemplar el caso en el que el **número de elementos que el usuario quiera almacenar sea superior al máximo valor de elementos por caja**. En esa circunstancia, el excedente del *Elemento* se añadirá en tantas cajas nuevas del *Almacen* como sea necesario hasta completar su almacenaje. [1.5 puntos]
 - Por ejemplo: si el usuario desea incorporar 16 elementos con la descripción “Paquetes de folios DINA4”, y la referencia 1255, el programa deberá registrar en la variable *Almacen* 3 cajas “llenas” (con 5 paquetes de folios cada una) y una cuarta caja con 1 único paquete de folios
- Antes de la finalización del programa se deberá **visualizar por terminal** toda la información almacenada en la variable *Almacen*. [1 punto]

Ejercicio 2 (5 puntos)

Escriba un programa en C++11, *Ejercicio2*, utilizando los tipos de datos, las **estructuras de control** y las **funciones** necesarias para procesar cada una de las frases (tipo `std::string`) que, de forma cíclica, se solicitará al usuario que teclee hasta que decida terminar el proceso.

La información que se tiene que obtener después de **analizar cada frase** se debe almacenar en **una sola variable**, del tipo que **consideres más adecuado**, con el siguiente contenido:

- **Número palabras** que forman la frase la frase.
- **Número de palabras que empiezan por vocal** en la frase.
- **Número de palabras que NO empiezan por vocal** en la frase.
- **Palabra de mayor longitud** en la frase. Si hay varias, se guardará la primera o la última de la frase.

El análisis de cada frase se hará **a través de una función que devolverá un puntero inteligente a la variable definida para almacenar la información obtenida en el proceso de análisis y descrita anteriormente [1 punto]**. Esta función puede contener llamadas a otras funciones que se consideren necesarias para realizar el análisis de la frase, es decir, para determinar:

- **Cuántas palabras** tiene la frase [**1 punto**].
- **Cuántas palabras empiezan por vocal y cuántas no** [**1 punto**].
- **Cuál es la palabra de mayor longitud** de la frase [**1 punto**].

Finalmente, realizar otra función que **reciba como argumentos la frase y el puntero inteligente a la variable donde se ha guardado todo lo calculado en el proceso de análisis** para mostrar por pantalla todo su contenido [**1 punto**].

Importante: se supone que la frase está bien escrita, es decir:

- Sólo contiene un espacio entre cada palabra.
- No hay espacios en blanco ni al principio ni al final de la frase.
- No existen signos de puntuación en la frase, solo palabras, y si existen, se toman como parte de las palabras.

Por ejemplo: para la frase “**Si no puedo dibujarlo es que no lo entiendo**”, deberá indicarse por pantalla lo siguiente:

La frase - **Si no puedo dibujarlo es que no lo entiendo** esta formada por:

Palabras: 9

Palabras que empiezan por vocal: 2

Palabras que NO empiezan por vocal: 7

Palabra de mayor longitud: dibujarlo