

ALUMNO:

EPS -Ingeniería Informática

Asignatura: G0460006 Programación I – TURNO 2

Curso: 2022/2023

Examen: Final

Fecha: 26-6-2023 (11:15-14:15)

Semestre: 1º

Convocatoria: Extraordinaria

### Parte Práctica (10 puntos; 70% nota final) - Tiempo: 2h40m

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

<b>Funciones/Métodos:</b> Si no se usa el paso por referencia constante cuando las variables de los parámetros de entrada no son de tipo simple.	40%
<b>Tipos de datos y variables:</b> <ul style="list-style-type: none"><li>• Uso de variables globales (fuera del ámbito de una función).</li><li>• 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.</li><li>• Si no se usan punteros inteligentes (std::unique_ptr; std::shared_ptr) cuando sea necesario</li></ul>	0% 0% 0%
El <b>programa no compila</b> o <b>no se asemeja</b> a lo pedido.	0%
Si no se cumplen los <b>criterios de entrega</b> indicados en la <b>actividad/examen</b> .	0%

#### Criterios particulares de evaluación

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%

#### 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.
- Todos los ejercicios resueltos sin utilizar funciones cuando sea apropiado se valorarán con una nota máxima del 60% sobre la calificación prevista.
- 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 (3 puntos)**

Escriba un programa en C++11, *Ejercicio1*, utilizando los tipos de datos, las **estructuras de control** y las **funciones** necesarias para **capturar de forma cíclica**, hasta que el usuario desee finalizar el programa, **dos números enteros mayores que cero desde el teclado**.

El programa debe cumplir los siguientes requisitos:

- Los valores capturados se almacenarán en **dos variables** de tipo **int**, cuyas etiquetas serán *numero* y *posición*. Si los valores no cumplen los requisitos indicados anteriormente se mostrará por terminal un mensaje de advertencia al usuario. **[0,5 puntos]**
- Si los valores cumplen los requisitos el programa **mostrará por terminal el dígito del número indicado por la posición cuando esté dentro del rango**. Si la **posición excede el número de dígitos del número introducido por teclado** el programa mostrará por terminal el mensaje “**Posición fuera de rango**”. **[2 puntos]**
- El programa debe **realizar su tarea de forma cíclica hasta que el usuario decida terminar** el programa. **[0,5 puntos]**

De manera que si los valores introducidos por teclado para **número** y **posición** son respectivamente:

- 1024 y 3, el programa mostrará por terminal el valor 4
- 1024 y 0, el programa mostrará por terminal el valor 1
- 1024 y 4, el programa mostrará por terminal el mensaje “Índice fuera de rango”

## **Ejercicio 2 (3 puntos)**

El registrado de un sistema de control de inmersión de un submarino almacena las siguientes **medidas de profundidad** junto al **tiempo invertido en alcanzar dicha profundidad**.

<b>Profundidad [m]</b>	0	5	15	8	24	13	4	4	0
<b>Tiempo [h y m]</b>	0h1m	0h5m	0h10m	0h14m	0h45m	1h10m	1h15m	3h0m	3h4m

Escriba un programa en C++11, *Ejercicio2*, utilizando los tipos de datos, las **estructuras de control** y las **funciones** necesarias para que cumpla los siguientes requisitos:

- En la función principal **declaréis e inicialicéis una única variable de tipo contenedor** que almacene la **información almacenada en la tabla anterior** **[0,5 puntos]**
- Implementar una función que devuelva para el registro almacenado la **secuencia de inmersiones y emersiones** que ha realizado durante las maniobras realizadas, mediante un **puntero inteligente a una variable contenedor** del tipo que consideres oportuno que contenga las acciones realizadas durante la maniobra del submarino. Dichas maniobras serán “**inmersion**”, “**emersion**” y “**estable**” según aumente la profundidad, disminuya o no varíe respectivamente. **[1.25 puntos]**
  - Si la función en lugar de devolver un puntero inteligente devuelve únicamente una variable contenedor este apartado puntuará como máximo **[0,75 puntos]** en lugar de **[1.25 puntos]**
- Implementar una función que devuelva para el registro almacenado las **operaciones realizadas** (inmersiones, emersiones y estable) junto la **velocidad media** [m/hora] **de cada una de ellas**. Esta función debe utilizar la función del apartado anterior para la obtención de las maniobras realizadas por el submarino. **[1.25 puntos]**
  - En caso de no haber implementado la versión con punteros este apartado puntuará como máximo **[0,75 puntos]** en lugar de **[1.25 puntos]**

**Para puntuar los diferentes apartados de este ejercicio debes probarlos en el main.**

**No se tendrá en cuenta para la puntuación de este ejercicio los apartados no probados.**



### **Ejercicio 3 (4 puntos)**

Escriba un programa en C++11, *Ejercicio3*, utilizando los **tipos de datos**, las **estructuras de control** y las **funciones** necesarias para capturar **un texto cualquiera** desde el teclado de **manera cíclica** hasta que el usuario quiera finalizar el programa [**0,5 puntos**]. Además, dicho programa debe incorporar:

- Una función de etiqueta ***cambiar*** que busque un **patrón** pasado por argumento dentro de la **cadena de texto capturada** y lo sustituya por la **cadena de texto de sustitución** que la función también recibirá como argumento.
  - La cadena de texto inicial no debe ser modificada y la sustitución debe ser realizada en todas las coincidencias.
  - Por ejemplo, si la cadena capturada fuera “**Hola Pepe. Quieres ver la película?**”, el patrón a buscar “**pe**” y la cadena de sustitución “**yYyYy**”. El retorno de la función debería ser “**Hola PeyYyYy. Quieres ver la yYyYylicula?**” [**1,5 puntos**]
- Una función de etiqueta ***cifrar*** que a partir de la **cadena de texto** capturada desde el teclado y un **valor entero** devuelva el resultado de aplicar el método de cifrado César. Dicho cifrado codifica un texto desplazando cada uno de los caracteres un número entero de posiciones en el abecedario. De tal manera que si el texto original fuera “**ABCDEFGHIJKLMNOPQRSTUVWXYZ**” y el desplazamiento seleccionado para la codificación fuera **4**, el texto cifrado sería “**EFGHIJKLMNOPQRSTUVWXYZABCD**” [**2 puntos**]
  - En el abecedario que utilizaremos en el programa no consideraremos la letra Ñ.
  - Vuestra función debe ser compatible con **letras mayúsculas y minúsculas**.
  - En caso de que el la cadena de texto contenga **caracteres diferentes a las letras del abecedario** (como **espacios, signos de puntuación**, etc.) se dejarán tal cual, **sin codificar**.
  - Por ejemplo, el texto “**Hola! Vamos?**” con el desplazamiento **5** dará como resultado “**Mtqf! Afrtx?**”

**Para puntuar los diferentes apartados de este ejercicio debes probarlos en el main.**

**No se tendrá en cuenta para la puntuación de este ejercicio los apartados no probados.**