

ASIGNATURA: Lenguajes de Programación
PERÍODO ACADÉMICO: 2024-2
FECHA: 25/10/2024
TIEMPO: 120 minutos

NOTA

Laboratorio 2

CÓDIGO	APELLIDOS Y NOMBRES	SECCIÓN

ANTES DE INICIAR LA EVALUACIÓN DEBE LEER LAS INSTRUCCIONES

Si la evaluación indica cargar algún archivo en la computadora, recuerde que es responsabilidad del estudiante hacerlo en el tiempo establecido y con las instrucciones dadas. Debe indicar el número de la misma en el recuadro siguiente:

INSTRUCCIONES GENERALES:

- La evaluación consta de 4 preguntas. Solo podrá elegir una pregunta 0 (la A o la B), por lo que solo se evaluará una de ellas.
- Debe entregar TODO su código fuente por la plataforma Blackboard.
- Puede utilizar apuntes digitales (no físicos). **No debe de tener internet activado.**
- **Leer detenidamente las situaciones que ocasionarán la anulación de la evaluación, que se encuentran a continuación.**

SITUACIONES QUE OCASIONARÁN LA ANULACIÓN DE LA EVALUACIÓN:

- Mantener prendidos teléfonos celulares, relojes smart, así como cualquier otro medio o dispositivo electrónico de comunicación.
- Utilizar material de consulta no autorizado (apuntes de clase, fotocopias o materiales similares).
- Utilizar calculadora no estando permitido.
- Compartir o intercambiar hojas, tablas, cualquier material impreso, dispositivo electrónico, durante el desarrollo de la evaluación.
- Conversar durante el desarrollo de la evaluación.

Los profesores de la asignatura

<div></div> <div>FIRMA DEL ALUMNO (LEYÓ LAS INSTRUCCIONES)</div>
--

PREGUNTA 0.A (4 puntos)

Se le pide implementar una clase llamada Personaje, que tenga 3 propiedades:

- nombre, que sea de tipo string y que sea privada.
- salud, que sea de tipo float y que sea pública.
- RecibirDanho, que sea un método público.
- Constructor que permita recibir un nombre del personaje y setee la propiedad salud en 100.

Los objetos que se instancian de esta clase implementada pueden recibir el mensaje RecibirDanho, y lo que tienen que hacer como reacción ante este mensaje, es disminuir en uno la propiedad salud.

Nota: Todo el código debe encontrarse en una unidad de compilación llamada p0

PREGUNTA 0.B (4 puntos)

Se tiene una lista de tuplas que representan distintos cursos en la Universidad de Lima, así como información de estos, tal como se muestra el siguiente ejemplo:

```
listaCursos :: [(String, String, String, Int, Float)] = [  
    ("Sistemas", "Ingeniería de Software", "Lenguajes de Programación", 40, 12.34),  
    ("Sistemas", "Ingeniería de Software", "EDA", 38, 40.34),  
    ("Industrial", "Investigación de Operaciones", "Simulación de Procesos", 42, 30.34),  
    ("Industrial", "Operaciones y Cadena de Suministros", "Gestión de la Demanda", 40,  
12.34),  
    ("Civil", "Construcción", "Tecnología de Construcción I", 42, 15.77)  
    ...]
```

Como se nota en el ejemplo, cada tupla está formada de 5 elementos:

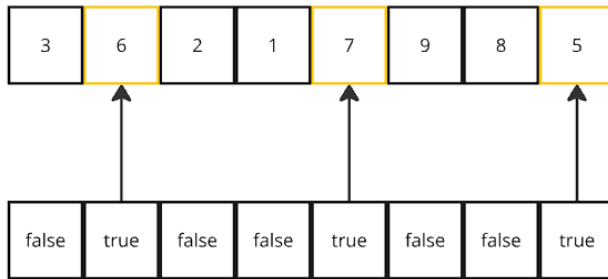
- Nombre de la carrera (String)
- Nombre del área académica (String)
- Nombre del curso (String)
- Número de matriculados en el curso (Int)
- Porcentaje de desaprobados (Float)

Implementar una función llamada alumnosMatriculados que reciba como argumento de entrada la lista de cursos y el nombre de la carrera, y devuelva la cantidad total de alumnos matriculados (Int) en todos los cursos de determinada carrera.

NOTA: Debe obligatoriamente utilizar map y/o filter (otra manera se calificará con 0). Llamar al archivo p

PREGUNTA 1 (4 puntos)

Se le pide implementar una función llamada **SumarSi** que reciba dos arrays, uno de 8 enteros y el otro de 8 booleanos. La función debe calcular la suma de los números en la posiciones que son true en el arreglo de booleanos, tal como se muestra en el siguiente ejemplo:



Se sumaría: $6 + 7 + 5 = 18$

Nota: Todo el código debe encontrarse en una unidad de compilación llamada p1.

PREGUNTA 2 (6 puntos)

Se le pide implementar una aplicación que ayude a registrar información de tarjeta de pokemones, para esto, se le proporciona la siguiente cabecera:

```
struct Pokemon
{
    std::string nombre;
    float health;
    float attack;
    float defense;
};

std::vector<Pokemon> ObtenerPokedex();
void ImprimirPokedex(std::vector<Pokemon> pokedex);
```

Debe implementar la función **ObtenerPokedex (2 puntos)** que se encargue de preguntar al usuario (por medio de la entrada estándar) la cantidad de pokemones que desea agregar al pokedex, y según este dato, le vaya pidiendo uno por uno la información de cada pokemon (nombre, health, attack y defense) y agregándola en una estructura de datos vector, para luego al terminar la ejecución de la función devolverla. Además, se le pide implementar la función **ImprimirPokedex (4 puntos)**, que debe recibir la estructura de datos vector y en base a ello imprimir la siguiente información:

```
Nombre: Pikachu
Salud: 100
Ataque: 40
Defensa: 30
Nombre: Bulbasaur
Salud: 100
Ataque: 50
Defensa: 30
Nombre: Charizard
Salud: 100
Ataque: 25
Defensa: 45
Nombre Pokemon Mayor: Charizard
Nombre Pokemon Menor: Pikachu
```

Como ve, se debe imprimir la información de cada Pokemon y al final imprimir el nombre del Pokemon que tiene mejor promedio entre ataque y defensa.

Nota: Todo el código debe encontrarse en una unidad de compilación llamada p2.

PREGUNTA 3 (6 puntos)

Usted quiere realizar una inversión en un proyecto de tecnología que le asegura tener rédito luego de una cantidad de meses (interés). Para esto debe de solicitar al usuario la siguiente información (por la entrada estándar o línea de comandos): el monto capital de la inversión (float), el interés mensual (float) y el número de meses (int) en las que desea obtener el retorno. Por lo tanto, se le pide implementar una función llamada **ObtenerDatos (2 puntos)** que pida los datos anteriormente detallados y los devuelva como una tupla. (std::tuple). Además, debe implementar la función llamada **ImprimirReporte (4 puntos)**, que reciba como argumento de entrada la tupla e imprima en pantalla la siguiente información de la inversión realizada:

Para una inversión de S/. 1000, con un interés compuesto de 10% y una cantidad de 5 meses.

```
=====
Mes: 1
Capital: 1000
Interes: 100
Nuevo Capital: 1100
=====
=====
Mes: 2
Capital: 1100
Interes: 110
Nuevo Capital: 1210
=====
=====
Mes: 3
Capital: 1210
Interes: 121
Nuevo Capital: 1331
=====
=====
Mes: 4
Capital: 1331
Interes: 133.1
Nuevo Capital: 1464.1
=====
=====
Mes: 5
Capital: 1464.1
Interes: 146.41
Nuevo Capital: 1610.51
=====
El interes total por recibir es de : 610.51
```

Como ve, al interés compuesto, este se acumula al capital para el cálculo del siguiente mes. El interés total por recibir es el capital final menos el capital inicial.

Nota: Todo el código debe encontrarse en una unidad de compilación llamada p3.