

Bienvenidos al primer parcial de Programación Avanzada de la facultad de ingeniería del departamento de ingeniería de sistemas de la Pontificia Universidad Javeriana. De acuerdo con el numeral 123 del reglamento de estudiantes, constituye una falta grave "el fraude en actividades, trabajos y evaluaciones académicos..." Por lo tanto, **todos los casos de fraude identificados serán reportados.**

No se pueden utilizar dispositivos electrónicos (Computadores, Celulares, Tablets u otros dispositivos).

Nombre: \_\_\_\_\_

ID: \_\_\_\_\_

### I. (20 PUNTOS) CONCEPTOS

- 1) Cuando se pasa un arreglo como parámetro a una función, éste se pasa como una copia
  - a. Verdadero
  - b. Falso
- 2) ¿Qué hará el siguiente código?

```
1 int arr[5] = {1, 2, 3, 4, 5};
2 int *ptr = arr;
3 *(ptr + 2) = 10;
```

  - a. Asignará el valor 10 al primer elemento de arr.
  - b. Asignará el valor 10 al tercer elemento de arr.
  - c. Asignará el valor 10 al segundo elemento de arr.
  - d. Asignará el valor 10 al cuarto elemento de arr.
- 3) ¿Cuál es el formato correcto para abrir un archivo de texto en modo de escritura en C++?
  - a. ofstream file("archivo.txt", ios::in);
  - b. ifstream file("archivo.txt", ios::out);
  - c. ofstream file("archivo.txt", ios::out);
  - d. fstream file("archivo.txt", ios::in);
- 4) En C++, el operador new se utiliza para liberar memoria asignada dinámicamente.
  - a. Verdadero
  - b. Falso
- 5) ¿Cuál de las siguientes sentencias no es correcta?
  - a. char frase [] = "Esta es una frase";
  - b. char \* frase2 = "Esta es una frase";
  - c. string frase3 = "Esta es una frase";
  - d. string \* frase4 = "Esta es una frase";
- 6) ¿Cuál de las siguientes notaciones es correcta para acceder al nombre de la estructura persona utilizando el apuntador perPtr?

```
1 struct Persona {
2     string nombre;
3     int edad;
4     float estatura;
5 };
6
7 int main() {
8     Persona persona;
9     Persona * perPtr = &persona;
10    return 0;
11 }
```

- a. perPtr.nombre
  - b. perPtr->nombre
  - c. \*(perPtr->nombre)
  - d. \*(perPtr)->nombre
- 7) ¿En C++ al abrir un archivo solamente se puede abrir como escritura o lectura, pero no ambos?
    - a. Verdadero
    - b. Falso
  - 8) ¿Qué función de C++ se utiliza para asignar memoria dinámica para un arreglo?
    - a. malloc
    - b. calloc
    - c. new
    - d. free
  - 9) Un struct en C++ puede contener tanto datos como funciones.
    - a. Verdadero
    - b. Falso
  - 10) ¿Cuál es la forma correcta de liberar memoria asignada dinámicamente para un arreglo en C++?
    - a. delete p;
    - b. free(p);
    - c. delete p[];
    - d. delete[] p;

### II. (80 PUNTOS) GESTIÓN DE RESERVAS

Una compañía aérea necesita desarrollar un sistema básico para gestionar aviones, vuelos y pasajeros. Como desarrollador, la tarea a su cargo es implementar un conjunto de funcionalidades que permitan gestionar esta información.

#### Información de un pasajero:

- 1) **Nombre:** cadena de caracteres de 50 bytes.
- 2) **Apellido:** cadena de caracteres de 50 bytes.
- 3) **Documento:** entero (int).



- 4) **Nacionalidad:** cadena de caracteres de 30 bytes.

**Información de un asiento:**

- 1) **Número de asiento:** entero (int).
- 2) **Ocupado:** booleano; false si está vacío y true si está ocupado.
- 3) **Pasajero:** estructura Pasajero; si hay un pasajero, en este campo se almacenan sus datos.

**Información de un vuelo:**

- 1) **Código de vuelo:** cadena de caracteres de 10 bytes.
- 2) **Origen:** cadena de caracteres de 50 bytes.
- 3) **Destino:** cadena de caracteres de 50 bytes.
- 4) **Fecha:** cadena de caracteres de 11 bytes en formato DD/MM/AAAA.
- 5) **Asientos:** arreglo dinámico; la cantidad de asientos varía según el avión.

**Información de un avión:**

- 1) **Nombre de la empresa:** cadena de caracteres de 50 bytes.
- 2) **Código del avión:** cadena de caracteres de 10 bytes.
- 3) **Número de asientos:** entero (int).
- 4) **Vuelos:** arreglo dinámico, la cantidad de vuelos varía.

**Información de la compañía:**

- 1) **Aviones:** arreglo dinámico; la compañía puede agregar más aviones a una empresa específica.
- 2) **Número de aviones:** entero (int).

La compañía aérea ya cuenta con un archivo texto llamado **vuelos.txt** que contiene información sobre vuelos y aviones. Un ejemplo del archivo es el siguiente:

```
Avianca
6052-20,6053-30
6052,AV1120,Bogota,Rio de Janeiro,21/11/2024
6052,AV1121,Rio de Janeiro,Bogota,12/12/2024
6053,AV1240,Bogota,Cali,13/10/2024
6053,AV1242,Medellin,Bucaramanga,10/02/2025
#
```

```
American Airlines
2040-25
2040,AA1120,Bogota,Miami,10/05/2024
2040,AA1124,Bogota,Miami,19/05/2022
2040,AA1121,Miami,Madrid,12/12/2024
#
```

Por ejemplo, en la primera línea del archivo se encuentra el nombre de la empresa (Avianca). En la segunda línea se especifican los códigos y el número

de asientos de cada avión, separados por el carácter "-" (si hay más de un avión, se separa por comas; por ejemplo, Avianca tiene dos aviones, el código del primero es 6052 con 20 asientos y el del segundo es 6053 con 30 asientos). Finalmente, en las demás líneas se encuentran los vuelos. El primer valor es el código del avión (6052), seguido por el código del vuelo (AV1120), el origen (Bogota), el destino (Rio de Janeiro) y la fecha (21/11/2024). El carácter "#" indica que ya no hay más vuelos asignados a la empresa y que se pasa a la siguiente.

A partir de la información proporcionada, se le solicita implementar los siguientes servicios:

- 1) **(5 puntos) Definir la estructura de pasajero y asiento:** escribir la definición de la estructura de pasajero y asiento.
- 2) **(15 puntos) Agregar un avión a la compañía:** desarrollar una función que permita agregar un avión a la compañía, en la que se envíe como parámetro la compañía por referencia y el avión por valor.
- 3) **(5 puntos) Buscar avión por código:** desarrollar una función que, dado el código de un avión, lo busque en el arreglo de aviones de la compañía y lo retorne por referencia. Si no lo encuentra, debe retornar **nullptr**.
- 4) **(20 puntos) Agregar un vuelo a un avión:** desarrollar una función que, a partir de una cadena de caracteres (línea del vuelo en el archivo de texto), busque el avión con el método anterior y agregue el vuelo al arreglo de vuelos del avión.
- 5) **(20 puntos) Leer el archivo de texto:** desarrollar una función que lea un archivo de texto llamado **vuelos.txt**, utilizando los métodos anteriores para agregar aviones y vuelos registrados en la compañía.
- 6) **(15 puntos) Generar reporte de vuelos por ciudad de origen:** dada una cadena de caracteres, buscar los vuelos cuyo origen coincida con dicha cadena. Si se encuentran vuelos, generar un archivo de texto llamado **vuelosOrigen.txt** y un archivo binario llamado **vuelosOrigen.dat** con la información de los vuelos.

**Restricciones:** no se pueden utilizar datos de tipo **string**, los arreglos deben ser dinámicos y deben recorrerse con aritmética de punteros.