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, **toda falta contra el reglamento de estudiantes seguirá el conducto regular hacia un posible proceso disciplinario.**

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

Nombre: _____

CC / TI: _____

Nombre del profesor _____

I. (20 PUNTOS) CONCEPTOS

1) Dado el siguiente código, ¿qué imprimirá?

```
1 int arr[] = {10, 20, 30, 40};
2 int *ptr = arr;
3 ptr++;
4 cout << *ptr;
```

a. 10
b. 20
c. Dirección de memoria de *arr[1]*
d. Error de compilación

2) En C++, cuando declaramos un apuntador y reservamos memoria con el operador new, la memoria se libera automáticamente cuando el programa termina.

a. verdadero
b. falso

3) ¿Qué resultado produce el siguiente código?

```
1 char str[] = "Hola";
2 std::cout << *(str + 2);
```

a. *H*
b. *o*
c. *l*
d. *a*

4) Para leer datos de un archivo binario en C++, se utiliza:

a. `read()`
b. `get()`
c. `getline()`
d. `scanf()`

5) ¿Cuál es la diferencia entre *delete* y *delete[]* en C++?

a. *delete* se usa para liberar memoria de variables individuales, *delete[]* para arreglos dinámicos.
b. No hay diferencia, ambos se pueden usar indistintamente.
c. *delete* solo se usa con punteros a objetos, *delete[]* con punteros a estructuras.
d. *delete[]* libera más memoria que delete.

6) ¿Qué utilizaría preferentemente para almacenar información de estudiantes con nombres, apellidos y calificaciones?

a. Un arreglo simple
b. Una matriz
c. Un struct
d. Un apuntador

7) ¿Qué hace el siguiente código?

```
1 int a = 10;
2 int *p1 = &a;
3 int **p2 = &p1;
4 cout << **p2;
```

a. Imprime la dirección de *a*.
b. Imprime 10.
c. Lanza un error de compilación.
d. Imprime la dirección de *p1*.

8) En una función que recibe un parámetro por referencia:

a. La función recibe una copia del valor como parámetro
b. La función recibe la dirección de la variable que se envió como parámetro
c. La función puede modificar la dirección del apuntador
d. La función no puede acceder el dato del parámetro

9) ¿Qué operación se realiza en la siguiente línea de código?

```
1 ifstream archivo("datos.txt", ios::in);
```

a. Crear un nuevo archivo de texto para escritura
b. Abrir un archivo de texto existente para lectura
c. Abrir un archivo binario para lectura
d. Crear un nuevo archivo binario para escritura y lectura

10) ¿Qué hace la función write en un archivo binario?

a. Escribe un apuntador en el archivo
b. Escribe una cadena de texto en el archivo
c. Escribe un número entero en el archivo
d. Escribe un bloque de memoria en el archivo

II. (80 PUNTOS) SISTEMA BANCARIO PARA GESTIÓN DE TARJETAS DE CRÉDITO

Un banco de Colombia desea desarrollar un programa en C++ para gestionar a los clientes y sus tarjetas de

crédito. El dueño del banco le ha proporcionado la siguiente información sobre las estructuras a utilizar:

Teléfono:

1) **numeroTelefonico:** cadena de caracteres de tamaño dinámico (14 bytes para celular, 11 bytes para teléfono fijo)

TarjetaCredito:

1) **numeroTarjeta:** cadena de caracteres de 17 bytes
2) **fechaVencimiento:** : cadena de caracteres de 6 bytes en formato (MM/AA)
3) **codigoSeguridad:** código de seguridad de la tarjeta (int)
4) **saldoDisponible:** saldo disponible en la tarjeta (double)

Titular:

1) **cedula:** cadena de caracteres que empieza por CC (20 bytes)
2) **nombreApellido:** cadena de caracteres de 40 bytes
3) **teléfonos:** arreglo dinámico de la estructura Telefono.
4) **cantidadTeléfonos:** número de teléfonos registrados
5) **tarjetasCredito:** arreglo dinámico de la estructura TarjetaCredito
6) **cantidadTarjetas:** número de tarjetas de crédito con las que cuenta el titular (int).

Banco:

1) **nombre:** cadena de caracteres de 40 bytes
2) **dirección:** cadena de caracteres de 45 bytes.
3) **titulares:** arreglo dinámico de la estructura Titular
4) **cantidadTitulares:** cantidad de titulares registrados en el banco (int).

El banco cuenta con el siguiente archivo de texto **cuentas.txt** en el que se encuentra la información existente del banco, de los titulares y de las tarjetas:

```
Banco de Bogota,Calle 36 #7-47
CC1234,Juan Fernandez,+573001526783,6014327865
4539876543211234,05/28,123,1500.75
5123456789012345,09/26,456,3500.00
CC543213,Maria Gomez,6014327865
5012345678901234,11/27,789,4200.50
#
```

1) La primera línea contiene la información del banco (nombre y dirección).
2) La segunda línea contiene la descripción del titular (cedula, nombre, apellido y varios teléfonos separados por comas)
3) Las líneas siguientes contienen las tarjetas de crédito que pertenecen al titular
4) Cuando aparece una nueva cédula, indica que un nuevo titular ha comenzado.
5) El carácter # significa que ha finalizado el archivo

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

1) **(5 puntos) Definir las estructuras Banco, Titular y Telefono:** escribir la definición de las estructuras Banco, Titular y Telefono.
2) **(20 puntos) Agregar teléfono titular:** dado un titular por referencia y un arreglo de caracteres con el número telefónico, adicionar el número al arreglo dinámico de teléfonos del titular. Nota: se debe validar que, si el número tiene prefijo, el arreglo de caracteres debe ser de 14 bytes (celular) y 11 bytes en caso contrario (fijo).
3) **(15 puntos) Agregar tarjeta crédito a titular:** dado el titular y la tarjeta de crédito por referencia, agregar la tarjeta al titular.
4) **(25 puntos) Cargar archivo de texto:** a partir del documento de texto *cuentas.txt* cargar los datos del banco, de las tarjetas, de los titulares y sus respectivos teléfonos en memoria.
5) **(10 puntos) Generar reporte Martes Visa (tarjetas que inicien con el número 4):** generar un archivo de texto que se llame *"martesVisa.txt"* de los titulares que cuentan con el descuento de martes Visa y con las tarjetas con las que puede aplicar. La función recibe como parámetro el Banco por referencia. En este archivo la información que debe mostrar tiene el siguiente formato en el que el carácter # es para separar al siguiente titular:

```
Nombre: Juan Fernandez
Celular: +573001526783, +573017895645,
Fijo:  6014327865,
4539876543211234, 05/28, 1500.75
#
Nombre: Maria Gomez
Celular:
Fijo:  6014327865,
#
```

6) **(5 puntos) Guardar tarjetas en archivo binario:** generar un archivo binario *tarjetas.dat* en el que se almacene solamente la información de las tarjetas de crédito de los titulares del banco. Esta función recibe como parámetro el banco por referencia.

Restricciones: no se pueden utilizar datos de tipo **string**, los arreglos deben ser dinámicos y deben recorrerse con aritmética de punteros. **Nota:** asuma que ya se encuentra implementado el método *agregarTitularBanco (Banco * banco, Titular * titular)* que sirve para agregar a un titular al banco.