

**Práctica de Sobrecarga de Funciones y Operadores**

La resolución de los siguientes ejercicios, deberán ir acompañados de listado del programa debidamente documentado, y su disquette, estos se presentarán en **un sobre manila** que deberá estar abierto. Esta práctica se efectúa en grupos de mínimo 2 personas y máximo 5 personas.

Toda práctica que sea una copia, será dividirá entre el número de copias encontrado, asignándoles esta nota final.

1. **(3 puntos)** Cree la clase matriz con todas sus funciones miembro fundamentales para procesar la matriz(lectura de datos de filas y columnas, cargado de matriz, mostrar la matriz) y efectúe las siguientes operaciones:

- Genere la matriz caracol (La matriz es cuadrada).
- Rotar k-veces las filas de la matriz.
- Eliminar una columna X.
- Adicionar una columna X.
- Sobrecargue el operador + para sumar dos matrices cuadradas.
- Sobrecargue el operador \* para multiplicar dos matrices cuadradas.
- Sobrecargue el operador ++ para adicionar una columna X.
- Sobrecargue el operador – para eliminar la columna X.

No olvide incluir la sobrecarga de funciones, para **2 funciones miembro** (Ejemplo: datos-dimensión de la matriz, y genera – genera la matriz caracol).

2. **(1,5 puntos)** Implemente una clase cadena que contenga como dato una cadena de caracteres y las funciones miembro visualizar cadenas, leer cadenas y concatenar cadenas, esta última será implementada aplicando una sobrecarga de operador +, de tal modo que sume o concatene dos cadenas. También sobrecargue el operador ++, para codificar un mensaje, por ejemplo, si la cadena era: “clave”, esta cambia a “dfbwf”, el siguiente carácter, y la sobrecarga del operador --, que decodifique el mensaje. Siguiendo con el ejemplo: entra esta cadena: “dfbwf” y sale “clave”.

3. **(1 punto)**En la siguiente implementación para manipular números complejos, adicione una sobrecarga de función y sobrecargue el operador != que devuelva 1 si dos números complejos son distintos y 0 en caso contrario; el operador + para efectuar la suma de dos números complejos.

```
class complejo{
    float r, i;
public:
    complejo(float a=0.0, float b=0.0)
    {   r = a;    i = b; }
}
```

```
float real();
float imaginario();
complejo operator + (complejo a);
int operator != ();
};

float complejo::real()
{ return r;
}
//Implementa todas las funciones miembro
```

4. (0.5 puntos) ¿Qué está incorrecto en la siguiente definición de función operador? Fundamente su respuesta.

```
class computador{
public:
    int operador+(int a, int b) { return a+b; }
};
```

**Sugerencia:** Recuerde que + es un operador binario

5. (1 punto) Cree una clase fecha con datos enteros que contenga la fecha de un día (dd/mm/aaaa) de calendario y diseñe una función **operador <**, que determine si una fecha es mayor a otra cronológicamente. El operador debe devolver un valor 1 que significa verdadero ó 0 que significa falso.

**Fecha de Presentación:** 7 de mayo