EXAMEN DE LENGUAJES DE PROGRAMACIÓN II INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN CONVOCATORIA DE FEBRERO

PREGUNTA 1. (3 puntos)

```
class vector
{
    int *b;
    int size;
    public:
    int tam(void) {return size;}
};
```

Completar la anterior clase para que la siguiente función pueda ser compilada y ejecutada correctamente.

```
vector funcion(vector v1)
{
    vector v2;
    vector v3(v1.tam()); //Crea un vector del tamaño indicado y relleno a 0

    v2=v1;
    for (int i=0; i<v1.tam(); i++)
        v3[i]=v2[i]+v1[i];
    v1=v2+v3;
    cerr<<v1;
    return v1;
}
```

NOTA 1: **cerr** corresponde a la salida estándar de errores, que por lo general, está asignada a la pantalla, al igual que **cout**.

NOTA 2: En la suma de dos vectores, cuando la longitud de ambos es diferente, se considera que la longitud es igual, pero rellenado el menor a ceros.

PREGUNTA 2. (3,5 puntos)

Una empresa vende diversos artículos que pueden ser clasificados en dos categorías: A y B. Los datos que son interesantes para cualquier artículo (A, B o cualquier categoría que pudiera existir en el futuro) son su precio de venta, código y las unidades vendidas. Para cada artículo de la categoría A es necesario almacenar el coste de fabricación unitario de cada artículo. Para los artículos de la categoría B, además del coste unitario de fabricación, se debe guardar el coste unitario de transporte. Los costes pueden ser diferentes para cada artículo.

Para calcular el beneficio obtenido en cada artículo se restan los costes al dinero obtenido de la venta:

<u>Artículo de categoría A:</u>

Beneficio = (precio de venta - coste de fabricación) * unidades vendidas

<u>Artículo de categoría B:</u>

Beneficio = (precio de venta - coste de fabricación - coste de transporte) * unidades vendidas

<u>APARTADO A. (2 puntos)</u>

Define las siguientes 4 clases:

1. Artículo.
2. Artículo A.
3. Artículo B.
4. Ventas.

La clase ventas debe contener **una** lista de artículos de las categorías A y B. Se deben hacer las funciones necesarias para insertar artículos en la lista, así como los constructores y destructores que se crean necesarios.

<u>APARTADO B. (1,5 puntos)</u>

Realizar una función que por medio del polimorfismo en tiempo de ejecución recorra la lista de artículos calculando el beneficio obtenido en cada artículo y el beneficio total de todos los artículos.

PREGUNTA 3. (2 puntos)
Dada la siguiente clase:

```
class buffer
{
    int v[50];
    int tam;
    public:
    vector (void) {tam=0;}
};
```

Modifica la clase para que sea genérica en lugar de para enteros. Una vez hecho esto, añade una función que guarde el contenido del buffer en un fichero, y otra que cargue el contenido de un fichero en el buffer. Ambas funciones serán miembro de la clase y recibirán como parámetro el nombre del fichero. El miembro **tam** indica el número de elementos que contiene el buffer. Es decir, un buffer puede contener 50 elementos como máximo, pero puede tener menos. Del mismo modo, en el fichero que se cargue puede haber menos o más de 50 elementos.

NOTA: No se pueden añadir funciones **inline** a la clase, es decir, el código de las nuevas funciones que se definan estará fuera de la clase.

PREGUNTA 4. (1,5 puntos)
Responde brevemente a las siguientes preguntas:

a) ¿Qué son variables de clase e instancia? ¿Qué son funciones de clase? (0,9 puntos)
b) ¿Para qué se usa la palabra **friend** en C++? (0,3 puntos)
c) ¿Cuándo se define una herencia como virtual? (0,3 puntos)