

Se desea realizar una clase en c++ para utilizar en una aplicación de ventas para una máquina expendedora de ananás. Para eso, se tiene la siguiente definición de una clase fruta, que es genérica, de modo de poder ser utilizada en cualquier otra máquina expendedora (jugos exprimidos, de licuados, y de otras frutas, etc).

A saber:

```
class fruta
{
    private:
        char* nombreFruta;
        unsigned int pesoEnGramos;
        float precioPorKilo;
    public:
        // Constructores
        fruta(); //Por defecto.
        fruta(const char*, unsigned int, float); //Parametrizado
        fruta(const fruta&); //Copia
        ~fruta();

        // Metodos
        void setNombre(const char*);
        char* getNombre (void);
        void setPeso(unsigned int);
        unsigned int getPeso(void);
        void setPrecioKilo(float);
        float getPrecioKilo(void);
        float getPrecioFinal(void);

        // Sobrecarga de operadores.
        friend ostream& operator<< (ostream& , const fruta& );
};
```

Se solicita implementar los siguientes constructores:

[1.1] Constructor por defecto. Debe inicializar el nombre de la fruta a "No detallado", el peso de la fruta a 0, y el precio por kilogramo a 0.

[1.2] Constructor parametrizado. Recibe el nombre de la fruta, el peso de la misma y el precio por kilo. Si el nombre de la fruta tiene mas de 32 caracteres, debe truncarlo en 32.

[1.3] Constructor de copia. Copia los datos del objeto fruta pasado como argumento al objeto fruta que se está instanciando. En todos los casos, tiene que realizar correctamente el manejo de memoria dinámica. No asumir que hay memoria disponible. Chequear que el sistema operativo haya podido asignar la memoria solicitada.

[2] Realizar los siguientes métodos, cuyo comportamiento se puede deducir del nombre.

void setNombre (const char*) → Modifica el nombre de la fruta. Debe limitar a 32 caracteres.

char* getNombre (void) → Devuelve un puntero al nombre de la fruta.

void setPeso (unsigned int) → Modifica el peso de la fruta.

unsigned int getPeso (void) → Devuelve el peso de la fruta.

void setPrecioKilo (float) → Modifica el precio por kilo de esta fruta.

float getPrecioKilo (void) → Devuelve el precio por kilo de esta fruta.

float getPrecioFinal (void) → Devuelve el precio a cobrar (hace la cuenta).

[3] Implementar la sobrecarga del operador << de modo que al utilizarlo imprima en pantalla los datos del objeto que lo utiliza. Por ejemplo: si tenemos el siguiente fragmento de programa:

fruta A((const char*) "Anana", 1200, 250);

Se trataría de un Anana, que pesa 1,2 kilogramos, y como el precio por kilo es \$250, su precio total sería \$300. Con lo cual, al hacer

```
cout << A << endl;
```

El programa debería imprimir en pantalla:

Datos de la fruta:

Nombre: anana.

Peso [gramos]: 1200

Precio por kilo [\$]: 250

Precio de esta unidad [\$]: 300

[4] Modificar la clase fruta, agregando una variable estática privada que cuente la cantidad de objetos instanciados.

Modificar los constructores (todos ellos) de modo que incrementen estas variables cada vez que se instancia un nuevo objeto de tipo fruta. Incluir un método estático para poder mostrar en pantalla el valor de esta variable.

Implementar un main que muestre el uso de todos los items pedidos.