Algoritmos y Programación II

Trabajo Práctico 2:

“De la góndola al cliente”

Objetivo

Desarrollar un programa orientado a objetos que resuelva la compra de un cliente en un supermercado.

Datos de entrada

Habrá un archivo “productos.txt” en donde cada línea tendrá un producto con los datos separados por un espacio o tabulador. El formato será:
producto precio 0 / 1 stock

0 significa que no está en oferta, 1: tiene un 10% de descuento.

Ejemplo:

arroz	1450	0	25
leche	700	1	52
jamon	8.82	0	3500

…

Los productos no tienen acentos ni eñes, los precios son de tipo double, las unidades de stock, son enteras.

Además, hay otro archivo “compra.txt” con el siguiente formato:
producto cantidad

Ejemplo:

leche	2
queso	150

…

Requisitos

Leyendo el archivo productos, generar un vector dinámico, Gondola. Este vector comenzará con un tamaño de 5, en caso de llenarse, deben duplicar el tamaño. En caso de vaciarse en más de la mitad, deben achicar el tamaño del vector a la mitad.

Luego, en base al archivo compra, generan un nuevo vector, Chango, solo con los productos disponibles. Los que no se encuentren en stock, debe salir un mensaje “No hay stock del producto …”. Los que haya stock, pero no alcance a cubrir el total de la compra, debe mostrarse por pantalla “Del producto *[producto]* solo hay *[x]* unidades”.

Debe mostrar el vector Chango con el detalle de cada ítem:

producto	precio_unitario	cantidad	oferta (S/N)	precio_item

…
Total: $ …

Con la compra deben actualizar el stock de cada producto. Si un producto queda con stock 0, se elimina del vector Gondola.

Antes de cerrarse la aplicación, deben actualizar el archivo productos.

La aplicación debe estar completamente orientada a objetos.

Como mínimo tendrá las siguientes clases:

- Producto
- Gondola
- Chango

Gondola y Chango serán vectores dinámicos de objetos de clase Producto.

Consideraciones

- No se puede utilizar ninguna clase de STL.

Qué se evalúa

- Funcionalidad
- Buenas prácticas: código, modularización, comentarios, claridad
- Interfaz de usuario
- Manejo de POO – pre y poscondiciones
- Documentación
- Utilización de memoria dinámica

Normas de entrega

Deben entregar:

- Documentación
    - Diagrama de clases UML
    - Descripción de cada TDA, indicando las pre y poscondiciones de cada una de las operaciones.
- Código fuente
    - En los archivos .h también van las pre y poscondiciones.
- Archivo makefile (todas sus rutas deben ser relativas a la carpeta del proyecto).
- Agregar a la entrega un archivo productos.txt y otro compra.txt para testeo.

La fecha límite de entrega será el martes 30/1 a las 23hs 55'.