

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA

LENGUAJES DE PROGRAMACIÓN 1

1er. examen

(Primer Semestre 2013)

Indicaciones Generales:

Duración: 3 horas.

No se podrá usar material de consulta. No se ofrecerá asesoría en la parte teórica.

No se pueden emplear variables globales, tipos de datos struct, funciones de manipulación de archivos, clases ni objetos, como la clase string y las funciones que se encuentran en bibliotecas como stringstream. Tampoco se podrán emplear las funciones strtok, strdup, sscanf, sprintf ni fopen.

Las soluciones deberán desarrollarse bajo un estricto diseño descendente, por lo que NO SE CALIFICARÁN aquellos módulos que son llamados por otros que estén incompletos. Cada módulo no debe sobrepasar las 20 líneas.

La presentación, ortografía y gramática influirán en la calificación

Puntaje total: 20 puntos

Cuestionario:

PARTE TEÓRICA

Según el capítulo 6 (Classes and Data Abstraction) del libro "C++ How to program" de Deitel, responda las siguientes preguntas:

Pregunta 1 (2 puntos) Explique por qué es importante encapsular los datos en una aplicación. Dé un ejemplo diferente al que propone el autor.

Pregunta 2 (1 puntos) Explique la importancia de las funciones miembro set y get en la implementación de una clase.

Pregunta 3 (2 punto) Explique qué puede ocurrir si al implementar una función miembro, ésta devuelve una referencia a un dato miembro privado.

PARTE PRÁCTICA

Un almacén de productos electrodomésticos cuenta con un archivo de textos en el que se almacenas los diferentes productos que almacena, así como los movimientos de entrada y salida realizados en un periodo de tiempo, el archivo es similar al siguiente:

FG00345	Cámara fotográfica modelo FX1056	3567	699.50	} Código - Descripción - cantidad en stock - Precio unitario, separados por uno o más espacios en blanco
BQ83816	Televisor LCD 32 pulgadas	1214	2560.80	
...				
HO06874	Memoria 4 GB USB	5623	78.20	
VENTAS:				
KL35665	2345.67 S 12345.32 E	5890.00 E	687.9 S ...	} Código del artículo, montos de operación de ese artículo, tipos de operación (E: entrada, S: salida). Pueden repetirse los artículos
BQ83816	14356.82 E 12456.99 E	...		
MY44223	345.67 E 1245.09 S	456.78 S	25634.76 S ...	
KL35665	12345.83 S	3451.82 E	...	
...				

En la primera parte del archivo se encuentran los artículos almacenados, aquí los artículos no se repiten; la cantidad en stock se refiere a la cantidad total de unidades del artículo que tiene el almacén en un instante dado. En la segunda parte del archivo se encuentran las operaciones de entrada y salida de los diferentes artículos, cada línea empieza con el código de un artículo seguido de las operaciones hechas en un día del artículo. Como en esa parte están los registros de un período de tiempo, los artículos se pueden repetir varias veces. Los artículos no están ordenados, de igual manera, no necesariamente están todos los artículos, sólo se encuentran los que se han movido; las

cantidades dadas se refieren al monto de operación del artículo (precio unitario x cantidad). Cada línea puede tener un número diferente de valores

Se quiere que usted escriba la implementación de las funciones dadas en el siguiente programa:

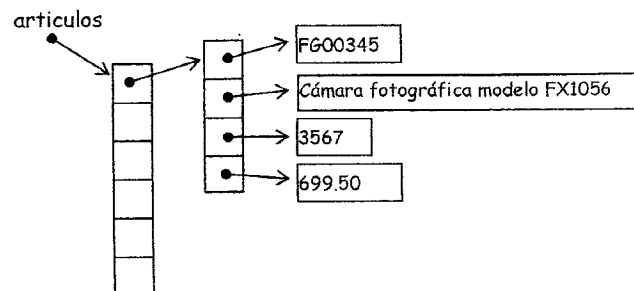
```
#include <funcs.h>
int main(void) {
    void *articulos, *movimientos;

    leeArticulos(articulos);
    leeMovimientos(movimientos, articulos);
    actualizaStock(articulos);
    eliminarArticulos(articulos, movimientos);
    reporteFinal(articulos);
    liberarEspacios(articulos, movimientos);
    return 0;
}
```

Pregunta 4 (4 puntos)

Función leerArticulos

Esta función deberá leer, re-direccionando la entrada estándar de datos, la primera parte del archivo descrito anteriormente y guardar la información en la estructura que se muestra a continuación:

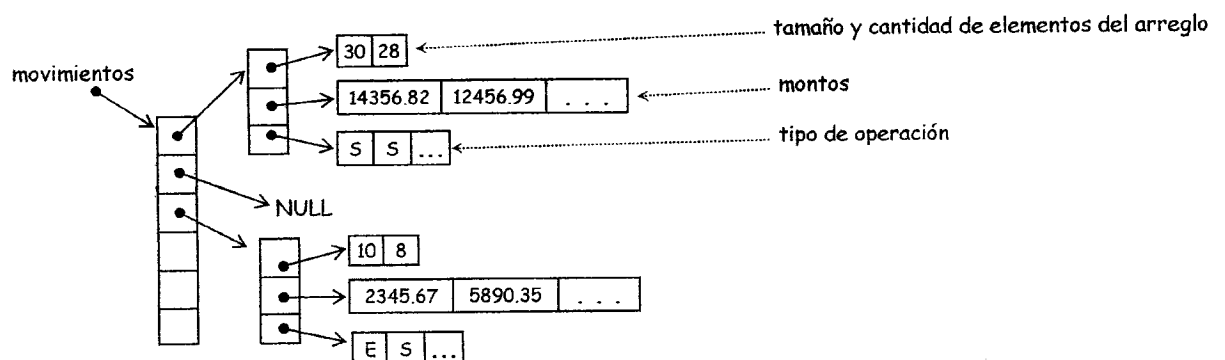


Los espacios de memoria deben ser exactos, los datos deben guardarse de acuerdo a su naturaleza, esto es las cantidades como enteros y los precios como valores de punto flotante.

Pregunta 5 (5 puntos)

Función leerMovimientos

La función debe leer la segunda parte del archivo y crear la estructura que se muestra a continuación, se deben colocar todos los datos del archivo sin resumir.



En la estructura, cada elemento del arreglo debe contener todos los movimientos realizados de un artículo, la posición de los datos deben coincidir con la de los artículos. El manejo de la memoria será dinámico; en el arreglo primario, todos los elementos empezarán apuntando necesariamente a NULL, luego los espacios de memoria irán creciendo de cinco en cinco. Si un artículo no tiene movimientos se quedará apuntando a NULL. El tamaño y cantidad de elementos se manejarán como enteros y los montos con valores de punto flotante.

Pregunta 6 (2 puntos)

Función **actualizaStock**

Esta función debe actualizar el stock de cada artículo según sus movimientos.

Pregunta 7 (2 puntos)

Función **eliminarArticulos**

Esta función debe eliminar de las estructura "artículos" y "movimientos", aquellos elementos que quedaron con un stock cero. La memoria de la estructura debe ser reasignada de modo de quedar siempre exacta.

Pregunta 8 (1 puntos)

Función **reporteFinal**

Esta función debe crear un archivo similar al de entrada de datos pero sólo con la primera parte.

Pregunta 9 (1 puntos)

Función **liberarEspacios**

Esta función debe liberar los espacios de memoria asignados a los estructuras.

Profesor del curso: Miguel Guanira

San Miguel, 14 de mayo del 2013