Programación y Estructuras de Datos (PED)

Examen sobre prácticas Junio 2020

Condiciones de entrega

- El examen se entrega a través del servidor de prácticas del DLSI http://pracdlsi.dlsi.ua.es**. Tras cada entrega, el servidor enviará al alumno un INFORME DE COMPILACIÓN**, para que el alumno compruebe que lo que ha entregado cumple las especificaciones pedidas y que se ha podido generar el ejecutable correctamente. **Este informe también se podrá consultar desde la página web de entrega de prácticas del DLSI** (http://pracdlsi.dlsi.ua.es *e introducir el nombre de usuario y password*). **SE ACONSEJA** que se haga una entrega previa con al menos el prototipo de la función pedida, para comprobar que no hayan errores de formato. Es posible que el servidor tarde más tiempo en devolver los informes en la última media hora del examen: **LO CUAL NO SERÁ ARGUMENTO** para justificar errores
- **En caso que la práctica esté correctamente entregada, compilada y ejecutada,** en este informe debe salir lo siguiente:

```
================================================
 DIFERENCIA CON FICHERO DE SALIDA DE REFERENCIA
================================================

----------------------------------------
```

- En el servidor, la práctica se comprobará con la siguiente función main:

```
TABBCom a; TListaCom L; TComplejo *x;
x = a.MostrarNiveles(L);
if (x==NULL) cout << "CORRECTO" << endl;
```

- Esperando la salida:

```
{}
CORRECTO
```

- Se tiene que entregar un fichero comprimido **tgz** (**tar cvzf fichero.tgz ***) que contenga todos los ficheros de los cuadernillos (con la estructura de directorios especificada en el enunciado de la práctica: **dentro del .tgz solo deben aparecer los directorios lib, include y src**), junto con los métodos pedidos en el examen. El examen debe compilar con todos los ficheros entregados.
- El fichero **nombres.txt** tiene que contener el nombre del único autor del examen.
- El nombre de la función implementada por el alumno debe coincidir EXACTAMENTE con el prototipo propuesto en el enunciado.
- El alumno tiene que implementar su propio fichero de prueba (**tad.cpp**) para comprobar el código implementado (**este fichero no es necesario entregarlo**).
- El alumno **puede añadir a la parte privada las variables y métodos** que considere necesarios para la implementación.
- **SI SE ENTREGA ALGO QUE NO COMPILA SUPONDRÁ UN CERO EN EL EXAMEN. Solo se evaluará la salida del programa. Se compilará con la versión del compilador instalada en los laboratorios de la EPS. Solo se evaluará la salida del programa.**
- **ARCHIVOS A ENTREGAR (incluyendo en el código la función MostrarNiveles) :**

```
include: tcomplejo.h, tvectorcom.h, tlistacom.h, tavlcom.h, tabbcom.h
lib: tcomplejo.cpp, tvectorcom.cpp, tlistacom.cpp, tavlcom.cpp, tabbcom.cpp
```

Añadir la siguiente función a la parte pública del TABBCom:

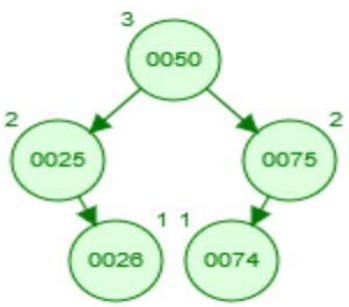
TComplejo* MostrarNiveles (TListaCom &)

La función **MostrarNiveles** devuelve un array dinámico de TComplejo y recibe una lista de TComplejo.
Hay que recorrer la lista (de izquierda a derecha) eliminando los elementos que ocupen las posiciones pares en la misma (se asume que el primer elemento de la lista ocupa la posición 1). La lista original se modifica y se devuelve por referencia.
A continuación, hay que copiar todos los elementos de la lista modificada (en la que se han eliminado los elementos de posiciones pares) en un array dinámico de TComplejo cuyo tamaño debe ser igual a la longitud de la lista modificada. Los elementos de la lista se recorren de izquierda a derecha y se insertan en el array de izquierda a derecha. Este array dinámico es devuelto por el método MostrarNiveles.
Posteriormente, hay que insertar todos los elementos del array dinámico obtenido previamente (recorriéndolo de izquierda a derecha) en el TABBCom que ha invocado al método MostrarNiveles. Si el TABBCom tenía elementos antes de la llamada a MostrarNiveles, todos los elementos deben ser eliminados previamente antes de las nuevas inserciones.
Por último, hay que recorrer el array dinámico (de izquierda a derecha) y para cada elemento del array hay que mostrar por pantalla, en líneas diferentes, el elemento y todos los elementos de su mismo nivel en el árbol (recorriendo todo el nivel de izquierda a derecha) con el siguiente formato: {Elemento: elemento. Elementos de su nivel: elemento1 elemento2}. Si el array estuviese vacío se mostraría {}. Para mostrar cada elemento TComplejo se utilizará el formato definido en el cuadernillo 1.

NOTAS:

- Si la lista de entrada está vacía se devolverá un array dinámico vacío.

Ejemplo (está hecho con números naturales por simplificación):



Ejemplo:

LISTA ENTRADA: [50, 23, 75, 42, 25, 68, 74, 83, 26]
ARRAY SALIDA: [50, 75, 25, 74, 26]
Mensajes por pantalla:
{Elemento: 50. Elementos de su nivel: 50}
{Elemento: 75. Elementos de su nivel: 25 75}
{Elemento: 25. Elementos de su nivel: 25 75}
{Elemento: 74. Elementos de su nivel: 26 74}
{Elemento: 26. Elementos de su nivel: 26 74}