



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Evaluación de Aprendizaje (EdA3).**Lea atentamente este documento antes de comenzar a resolver la ejercitación planteada.**

Debe resolver la siguiente problemática eligiendo la mejor estrategia.

Acerca de qué se trata el problema.

En una organización le encargan a un programador una función que permita determinar los **<nMejores>** calificados en una evaluación. Se le explica que esa evaluación puede tener puntajes o calificaciones (**<calif>**) del 0 al 10 ó cualquier otra escala de puntajes, que no serían precisamente calificaciones (podría ser el resultado de una evaluación de opciones múltiples o "*múltiple choice*" tomado mediante un "*form*" de "*Teams*").

Especificación funcional:

Los datos a procesar responden a un tipo de dato que contiene una clave (**<clave>** alfanumérica de 7 caracteres válidos que identifican al evaluado) y un puntaje (**<calif>** entero que refleja la calificación o puntaje obtenido). Se requiere una muy buena velocidad de procesamiento, por lo que se indica el uso de un TDA lista en que se eliminan los elementos cargados con anterioridad que ya no deben estar en la misma y luego se almacene el nuevo elemento de modo que queden ordenados por puntaje (de mayor a menor) y a igualdad de puntaje ordenados por la clave de menor a mayor (comparación lexicográfica).

Ejemplo: suponga que ya se han cargado algunos y que se buscan los cuatro mejores:

J-4 / P-4 / F-3 / J-2

Si ahora viniera: S-1 no se pondría en la lista porque ya hay cuatro mejores.

Si en cambio se trataba de: H-2 compartiría el 4to lugar con J-2

J-4 / P-4 / F-3 / H-2 / J-2

Si luego vinieran más datos con 2 compartirían el 4to lugar:

J-4 / P-4 / F-3 / A-2 / H-2 / I-2 / J-2 / M-2

Si viniera otro más con cualquier puntaje superior a 2, p.ej. C-3 su lugar de inserción estará antes de F-3 pero antes de insertarlo, se deben eliminar los que ya no pertenecen al grupo de los cuatro mejores quedando:

J-4 / P-4 / F-3

insertándolo en el lugar determinado antes:

J-4 / P-4 / C-3 / F-3



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Tipos de Datos Abstractos**Punto 1 a.-**

Escriba una primitiva `<ponerEnListaDobleNMejores>` del TDA-Lista implementada con memoria dinámica en una lista doblemente enlazada que resuelva el problema planteado. Las comparaciones necesarias para cumplir su propósito deberán hacerse invocando a la(s) función(es) que resuelvan la comparación. En este caso, no las recibe por argumento. Eso se hará en una segunda etapa por parte del equipo de la empresa que contrata su trabajo.

Las comparaciones deben respetar el hecho de devolver 0 (cero) si lo que se compara coincide. Algún valor negativo si al primero que interviene en la comparación le corresponde estar antes que al segundo. De otro modo un valor positivo.

Punto 1 b.-

Escriba una primitiva `<ponerEnListaSimpleNMejores>` del TDA-Lista implementada con memoria dinámica en una lista simplemente enlazada que resuelva el problema planteado. Valen las mismas consideraciones del punto anterior.

Ponga especial atención a lo pedido, es simple pero no es trivial.

Para la lista doblemente enlazada, tal como se vio en clase y apuntes, debe emplear puntero a nodo y en caso de insertar un nuevo nodo la variable `"lista"` de `"main"` debe quedar con la dirección del mismo (del último insertado).

De ningún modo debe utilizar puntero a nodo para recorrer la lista simplemente enlazada, tal como se vio en clase y apuntes de la materia.

Las funciones devolverán la cantidad de nodos eliminados (con un valor entero negativo), si no hubiera eliminación utilice la etiqueta **TODO_BIEN**, en caso de falta de memoria utilice **SIN_MEMORIA**. Se garantiza que no se dará clave duplicada por lo que no hay que tenerla en cuenta. Si no se puede insertar (porque ya hay cuatro -o n- mejores), deberá devolver con la etiqueta **NO_SE_INSERTA**.

NO se aceptará que recorra dos veces la lista, ni que inserte antes de eliminar nodos sobrantes (si los hubiera). Las listas NO se deben ordenar, se generan ordenadas tal como se indica al comienzo.

Debería considerar generar su propio lote de pruebas además del provisto con el proyecto asegurándose la correcta ejecución de las funciones pedidas.



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

POO-C++**Punto 2 a.-**

Declare con total corrección los métodos de las clases involucradas en el proyecto de instrumentos musicales demostrando su aprendizaje sobre polimorfismo y tratamiento de excepciones. El proyecto (la función "main") debe compilar sin errores ([Build] / [Compile current file]).

Punto 2 b.-

Desarrolle con total corrección los métodos de las clases involucradas en el proyecto de instrumentos musicales. El proyecto debe compilar sin errores y ejecutar sin violaciones de memoria.

Salida del proyecto de POO – C++

```
main.cpp [EaA3 - POO/C++ - Polimorfismo Instrumentos] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global>
Management
Projects Files FSymbols Resources
Workspace
EaA3 - POO/C++ - Polimorfismo Instrumentos
  source
    Cuerdas.cpp
    Instrumento.cpp
    Percusion.cpp
    Vientos.cpp
    main.cpp
  Headers
    include
      CambioEscalaException.h
      Cuerdas.h
      Instrumento.h
      Percusion.h
      Vientos.h
main.cpp x Instrumento.h x Instrumento.cpp x Cuerdas.h x Cuerdas.cpp x Percusion.h x Percusion.cpp x Vientos.h x Vientos.cpp x CambioEscalaException.h x
1 #include <iostream>
C:\Users\USER\Desktop\EdA3-2C\EdA3-2C-POO-C++RESUELTO\bin\Debug\EaA3_POO.exe
Esperado: Instrumento: Contrabajo - G
-----: Instrumento: Contrabajo - G
Esperado: Instrumento: Contrabajo - B
-----: Instrumento: Contrabajo - B
Esperado: Instrumento: Oboe - A material:madera
-----: Instrumento: Oboe - A material:madera
Esperado: Instrumento: Oboe - F material:madera
-----: Instrumento: Oboe - F material:madera
Esperado: Instrumento: Bateria - X
-----: Instrumento: Bateria - X
Esperado: Excepcion: No permite cambio de escala
-----: Excepcion: No permite cambio de escala
Process returned 0 (0x0)   execution time : 0.016 s
Press any key to continue.
```



UNLaM

Dto. Ingeniería e Investigaciones Tecnológicas

Condición mínima para no reprobar (4):

Punto 1 a.- o Punto 1 b.- además del Punto 2 a.-

Condición mínima para promocionar (7):

Punto 1 a.- y Punto 2 a.- además de Punto 1 b.- o Punto 2 b.-

Plazo de entrega: miércoles 02/12/2020 – 00:00 horas.