Universidad de Costa Rica
Escuela de Ciencias de la Computación e Informática
Semestre I - 2020
Curso CI-0113 - Programación II
Profesor: Edgar Casasola Murillo
Fecha: 10 de junio 2020

EXAMEN I

Para la resolución de este examen usted ya cuenta con la implementación de la Lista con Iterador (Esta Lista se utilizará en los ejercicios 2 y 3 por lo que se recomienda tener claro su funcionamiento e implementación). En el ejercicio 2 el método de inserción recibe por copia parámetros de tipo Lista::Iterador para obtener valores desde una Lista de este tipo.

1. (30%) Declare e implemente una clase llamada **ListaO,** que corresponden a una Lista Ordenada doblemente enlazada de valores con frecuencia y con Iterador (pueden basarse y reutilizar código del ejemplo de lista visto en clase) que permita **insertar valores** de tipo entero, **y acumular la frecuencia de cada valor**. En esta ListaO:
   a) Cada **valor** SIEMPRE se inserta dejando la lista *en orden de menor a mayor*.
   b) En cada celda existirá además un atributo llamado **frecuencia**.
   c) Al insertar un valor, si no existe en la ListaO se agregará una nueva Celda de forma que se respete el orden, y **si ya existe ese valor entonces se incrementará su frecuencia** de aparición.
   d) En el Iterador de esta lista el operador * solo debe retornar una copia del valor y no dar acceso. Por ejemplo: *iter = 3; Debe generar error al tratar de compilar el código.
   e) En el iterador debe existir además un método llamado int getFrecuencia() para retornar la frecuencia del valor al que apunta el iterador. Si el iterador es igual a Iterador::end() retorna 0.
   f) Se deben eliminar el acceso público a los métodos para agregar elementos al inicio o al final ya que dejarían la lista en forma inconsistente.

2. (30%) Suponga que se cuenta con grupos de datos (valores enteros) que corresponden a observaciones de grupos de datos asociados a un mismo valor específico. Se llamará Vecindario de un número n, a un objeto que almacena ese número, la cantidad de grupos donde fue visto y la lista de valores vecinos (con sus frecuencias).
Construya un objeto llamado **Vecindario** para almacenar **un valor entero** con su **frecuencia** de aparición y **las frecuencias de aparición de sus valores vecinos**. Además:
   a) El objeto debe contener como **<u>atributos</u>**: **el valor** asociado al Vecindario, la **frecuencia** de los contextos en los que fue observado ese valor, **y** por último **una lista de vecinos** con sus frecuencias.

b) Deberá tener **constructores** para recibir valores tanto desde vectores simples como desde Lista::Iterador.
   Un **constructor por omisión**. Lista vacía y valor y frecuencia en 0.
   Un **constructor que solo recibe el valor int**. Su lista de vecinos vacía.
   Un constructor que recibirá 3 parámetros.
   - Un int con el valor
   - Un Lista::Iterador (de la Lista vista en clase)
   - Un int indicando los N vecinos que se leerán a partir del iterador.

   Otro constructor que recibirá 3 parámetros.
   - Un int con el valor
   - Un parámetro puntero a entero: int * que apunta al vector de vecinos.
   - Un parámetro int N que indica la cantidad de vecinos

c) Deben existir **dos versiones del método insertar**. Una que reciba como parámetro:
   - Un Lista::Iterador que apunta al primero de una secuencia de vecinos.
   - Un int N que indica la cantidad de vecinos

   Y una versión para obtener los valores de los vecinos desde un vector:
   - Un parámetro puntero a entero: int * que apunta al vector de vecinos.
   - Un parámetro int N que indica la cantidad de vecinos

d) Se debe sobrecargar el operador de salida << para visualizar el contenido del objeto.

Por ejemplo: Al construir el Vecindario del número 06. Asuma que se cuenta con una secuencia de valores enteros donde aparece el número 06 luego de observar 5 números vecinos previos:
28 11 68 56 86 - 06 // aquí el valor 06 tiene como vecinos al 28 11 68 56 y 86
28 68 86 91 92 - 06
68 86 91 92 06 - 06 // note que un 06 puede tener a otro 06 como vecino
06 10 11 28 30 - 06
En este caso el **Vecindario del valor 06** contiene: **Frecuencia: 04** (porque son 4 contextos en los que se observó al número 06). Y el contexto del 06 contiene la siguiente **lista de valores vecinos y sus frecuencias:** *06 f:2 - 10 f:1 - 11 f:2 - 28 f:3 - 30 f:1 - 56 f:1 - 68 f:3 - 86 f:3 - 91 f:2 - 92 f:2*

3. Escriba una **aplicación de consola** para construir un programa que permita leer desde un archivo de texto una secuencia de valores enteros entre 0 y 99 y construya un vector con 100 Vecindarios (uno para cada número). Y luego al leer una secuencia de 5 números digitados por el usuario le permita recomendar un número.

Los vecinos de cada número serán los 5 valores previos a él observados en esta secuencia de valores. En otras palabras, si se tiene en el archivo de entrada los valores:

20 3 16 5 20 40 38 …

A partir del 40 se tiene una observación para crear el Vecindario del 40 con los valores 20 3 16 5 y 20. Así se obtiene el vecindario para el 40 f:1 vecinos: 3 f:1 - 5 f:1 - 16 f:1 - 20 f:2

Luego para el siguiente valor 38 su vecindario son los 5 valores previos
20 *3 16 5 20 40* 38 … así que se obtiene el vecindario del 38 f:1 vecinos: 3 f:1 - 5 f:1 - 16 f:1 - 20 f:1 - 40 f:1 … y así sucesivamente.

Luego de crear los 100 Vecindarios con esta información su programa le pedirá al usuario digitar 5 valores de un vecindario y deberá retornar el valor recomendado.

Fórmula para obtener el valor con el Vecindario más apropiado

El **valor entero recomendado** corresponderá al número cuyo vecindario tenga la **mayor suma de las frecuencias acumuladas para esos 5 valores digitados** por el usuario , **dividida luego entre la frecuencia del vecindario**.
En otras palabras, para cada Vecindario: se buscan y calcula la suma de las frecuencias de cada uno de los valores digitados por el usuario, y luego la suma se dividen entre la frecuencia del valor de ese vecindario particular. El valor del vecindario con mayor valor es el seleccionado.

Con los dos vecindarios anteriores si los números son:
Entrada de su programa: 06 28 20 00 05
La salida sería: 40
Ya que en el vecindario del 40 f:1 Vecinos: 5-f1 y 20-f2 con frecuencias 1 y 2 respectivamente y el 40 solo tiene 1 observación, así que nos dá por resultado: (1 + 2) / 1 mientras el 38 f:1 Vecinos: 5-f1 20-f1 nos da: (1+1) / 1

Y como 3 es mayor que 2 se selecciona el 40. En su caso tendrán más datos así que deberán evaluar todos los Vecindarios que no sean 0.

Pueden agregar a su clase Vecindario todos los métodos que consideren apropiados para resolver este problema si lo llegaran a necesitar.

FORMA DE ENTREGA:

Debe subir su solución al enlace respectivo en mediacionvirtual.ucr.ac.cr antes de las 11:55 p.m. del día Miércoles 10 de Junio del 2020.

Su solución puede consistir de:

1. Si hoy no cuenta con computador adecuado para la solución del examen (Casos especiales conocidos):
   Carpeta con Fotos o Documento en formato .pdf de su solución a puño y letra en caso de que no cuente con computador para trabajar el día de hoy. Explique su solución.

2. Carpeta con subcarpetas para cada pregunta. Debe incluir todo el código .h y .cpp y main de prueba con el que usted verificó el funcionamiento de las respuestas a cada pregunta.
   Puede replicar código en las carpetas con tal de que sea compilable sin problema.
   Incluya un archivo de texto llamado README.md donde explique como compilar, ejecutar cada solución y posibles problemas encontrados en caso de que no le compile.

PUNTOS EXTRA

C**omo parte de este examen** y en ambos casos se dará un 20% extra por entregar una solución extra debidamente documentada y funcional del examen. Cuya entrega será el martes 16 de junio por el medio y forma que se indicará en mediación virtual.