|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Universidad Tecnológica Nacional**  **Facultad Regional Avellaneda** | | | | | | | | | | |
| Técnico Superior en Programación  Técnico Superior en Sistemas Informáticos | | | | | | | | | | |
| Materia: Laboratorio de computación I | | | | | | | | | | |
| Apellido: |  | | | | Fecha: | | 27/06/2019 | | | |
| Nombre: |  | | | | Docente: | |  | | | |
| División: |  | | | | Nota: | |  | | | |
| Legajo: |  | | | | Firma: | |  | | | |
| Instancia | PP |  | RPP |  | SP | X | RSP |  | FIN |  |

**PARTE 1**

Un negocio de revelado digital de fotos posee un registro en un archivo de las ventas de revelados realizados por cliente y fecha. Realizar un programa que permita analizar esta información. El programa deberá iniciar y cargar en una lista los elementos definidos en el archivo data.csv, el cual cuenta con el siguiente formato:

**ID\_Venta;Fecha\_Venta;Tipo\_Foto;Cantidad;Precio\_Unitario;CUIT\_Cliente**

Por ejemplo:

**33;16/11/2018;4R\_10x15;30;10.99;20-33263109-4**

**34;17/11/2018;6R\_15x20;8;15.25;21-33782109-5**

**35;21/12/2018;POLAROID\_11x9;8;15.25;21-33782109-5**

Luego deberá generar un archivo “informes.txt” con el siguiente formato:

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Informe de ventas**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**- Cantidad de fotos reveladas totales: XX**

**- Cantidad de ventas por un monto mayor a $150: XX**

**- Cantidad de ventas por un monto mayor a $300: XX**

**- Cantidad de fotos polaroids reveladas: XX**

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

Los tipos de fotos son los siguientes:

* 4R\_10x15
* 5R\_13x18
* 6R\_15x20
* 8R\_20x25
* POLAROID\_11x9
* POLAROID\_10x10

Requerimientos del desarrollo.

* Se deberá crear la entidad “Venta” con todos sus campos correspondientes.
* Se deberá utilizar la biblioteca LinkedList para almacenar las ventas leídas del archivo.
* Se deberá agregar a la biblioteca la función “ll\_count()” la cual permitirá contar elementos de la lista bajo algún criterio.
* Se deberá utilizar la función “ll\_count()” para calcular los informes pedidos.

Detalle de la función “ll\_count()”

Prototipo de la función:

**int ll\_count(LinkedList\* this, int (\*fn)(void\* element))**

La función “ll\_count” recibirá una lista y una función “fn”. Se deberá iterar todos los elementos de la lista y pasárselos a la funcion “fn”. La función “fn” devolverá la cantidad que debe contarse. La función “ll\_count” almacenará un acumulador al cual sumará el valor de retorno de “fn” en cada iteración. Al finalizar las iteraciones, la función “ll\_count” devolverá el valor acumulado.

**Nota 0:** El código deberá tener comentarios con la documentación de cada una de las funciones y respetar las reglas de estilo de la cátedra.

**Nota 1:** Separar en archivos las entidades, parser y generador de informes.

**PARTE 2**

Preguntas oral:

* Estructura de la LinkedList.
* Función count.
* Otra función de la LinkedList.

**CONDICIONES DE APROBACIÓN**

Para la aprobación directa (nota >=6), se deberá tener el programa funcionando en su totalidad como se pide en la parte 1 del examen y haber contestado todas las preguntas de la parte 2.

Para la aprobación con final (nota = 4 o 5), se deberá realizar el parseo del archivo, la función ll\_count, las funciones para contar de al menos 2 informes y haber contestado la pregunta oral de la estructura de la LinkedList.