

**Taller #1**  
Período Lectivo 1-2016  
(Valor: 30%)

## 1. INSTRUCCIONES

- Este taller de programación es una evaluación estrictamente individual; por lo tanto, durante la realización del mismo, no está permitido prestar o recibir ayuda de otro(s) estudiante(s).
- ESTÁ permitido el uso de códigos fuentes en digital hechos por usted para el desarrollo del taller.
- Pueden traer material bibliográfico (libros, guías, manuales, código impreso, etc.) para el desarrollo del taller de programación.
- El estudiante que infrinja alguna de las instrucciones anteriores será retirado del taller y se le considerará aplazado con la nota mínima (Artículo 34 de las Normas de Evaluación de los Aprendizajes).
- **El taller tiene una duración estimada de tres (03) horas académicas.**
- El ejercicio propuesto requiere la lectura de datos de la entrada estándar (standard input) y la escritura de resultados en la salida estándar (standard output).
- Para el desarrollo del taller debe utilizar el lenguaje de programación C++, junto con sus librerías estándar. Para la compilación de sus códigos fuentes, debe realizarla por medio de un archivo makefile, el cual deberá entregar junto con sus códigos fuentes.
- Debe proveer un archivo **makefile** que permita compilar su taller, la falta del mismo conllevará a la no revisión de su taller, **adicionalmente dicho makefile debe generar un ejecutable con el nombre de "mediana"**, con el fin de facilitar las labores de corrección de su taller, el no hacerlo generará puntos menos en su nota final del taller.

## 2. ACTIVIDADES A REALIZAR

La **mediana** de una lista de enteros, se define como el elemento de dicha secuencia tal que la mitad de los elementos son menores que el y la otra mitad mayores o iguales. Suponiendo que dada una **lista  $L$  de enteros** no vacía, no ordenada y sin elementos repetidos, desarrolle un procedimiento que calcule la mediana de la lista  $L$  y retorne dos colas, una con los elementos menores a la mediana y otra con los elementos mayores o iguales que la mediana, ordenados ascendentemente, utilizando el paradigma de programación orientado a objetos. Para su implementación utilice la clase lista simplemente enlazada, y la clase cola vistas en el aula de clases, implementadas mediante el uso de plantillas.

### 2.1. Formato de entrada

La entrada constará de una primera línea con un entero  $n$  que indicará el número de casos de prueba. Cada caso de prueba constará de dos (2) líneas, en la primera línea se indicará un número entero  $m$ , que representará el número de elementos de la lista, la siguiente línea contendrá  $m$  números enteros separados por un espacio en blanco, los cuales serán cargados en dicha lista.

### 2.2. Formato de salida

La salida constará de  $n$  pares de líneas sin líneas vacías de por medio, que representarán la salida para los  $n$  casos de prueba, donde la primera línea contendrá los elementos menores a la mediana y la segunda línea los elementos mayores o iguales a la mediana, ordenados ascendentemente.

#### Ejemplo:

##### Entrada

```
2
5
30 10 40 20 50
6
300 100 600 500 200 400
```

##### Salida

```
10 20
30 40 50
100 200 300
400 500 600
```

### **3.OBSERVACIONES**

- No debe mostrar por pantalla ningún tipo de mensaje al usuario para su interacción con el programa, ninguna interfaz gráfica desarrollada por usted, será corregida positivamente o le aportara ningún punto extra para su calificación final
- El formato de salida debe ser respetado como se indica en el enunciado, de lo contrario acarreará con puntos menos en su nota final.