

1. Crear una aplicación en lenguaje Java que permita registrar la información relacionada con una báscula de camiones en una empresa de transportes. Para tal fin se desea almacenar los siguientes datos: placa del vehículo, peso (en toneladas), destino (1 = Buenaventura, 2 = Bogotá, 3 = Barranquilla), y fecha (día, mes y año) de salida del camión.

Los datos deben ser almacenados en estructuras de datos (arrays o matrices) de tamaño 50.

El día, mes y año se almacenan como datos separados.

Por ejemplo, suponga los siguientes datos:

Placa del vehículo: QWE123

Peso: 15

Destino: 1

Día: 24

Mes: 4

Año: 2020

La aplicación debe mostrar un menú en pantalla con opciones para:

- a. Ingresar los datos del pesaje
- b. Mostrar la cantidad de viajes realizados en total y el peso
- c. Mostrar la cantidad a los tres destinos y el peso promedio de los tres destinos
- d. Consultar los viajes por fecha y destino
- e. Salir

Para la opción a, solo se solicita el ingreso de un viaje. La aplicación debe tener un mecanismo de validación que permita determinar si es posible el almacenamiento de datos en las estructuras utilizadas.

Las opciones b, c y d, pueden ser ejecutadas en cualquier momento. No es necesario que las estructuras de datos estén totalmente llenas.

(2.0 pts.)

2. Llenar aleatoriamente una matriz de tamaño N x M con letras las 27 letras del abecedario más el espacio en blanco, es decir 28 caracteres en total. La matriz debe ser declarada de tipo char. La aplicación permite que el usuario ingrese la cantidad de filas y columnas de la matriz, pero debe validarse que la multiplicación de filas x columnas debe ser 28. Por ejemplo, 7 filas x 4 columnas, o 14 filas x 2 columnas, Etc. El usuario también puede escoger si la matriz se llena con letras mayúsculas o minúsculas. Mostrar la matriz por pantalla.

(1.0 pts.)

3. De los siguientes métodos de ordenamiento seleccionar tres de ellos,

- Inserccìon
- bucket sort o bin sort
- Counting sort
- Merge sort
- Radix sort
- Shell sort
- Seleccìon
- HeadSort
- Quicksort

realizar pruebas de ordenamiento y desempeño con tamaños de array de:

- 10.000
- 50.000
- 100.000
- 500.000
- 1.000.000

Tomar los datos de medición correspondiente (para cada algoritmo): tiempo de ejecución, cantidad de intercambios, cantidad de asignaciones y cantidad de comparaciones.

Se debe asegurar que las pruebas se realizan sobre el mismo array. Por ejemplo:

$A[] = \{789, 139, 50, 258, 8\}$

Quiere decir que los tres métodos seleccionados deben ordenar $A[]$. Para ello, utilice un array $B[]$ donde copie los valores originales de $A[]$, y cada que utilice un método de ordenamiento, restaure los valores de $A[]$ desde $B[]$.

(2.0 pts.)

Observaciones:

- Fecha máxima de entrega: 03 de febrero de 2021 06:00 am
- Método de entrega: enlace campus virtual
- Metodología: grupos de dos estudiantes