Department of Computer Science and Engineering
Green University of Bangladesh
CSE 205 - Data Structure

Homework 2 Marks: 10
Due: October 06, 2024
Note: Late projects will be assessed a 20% penalty for each day past the due date. Projects will not be accepted more than four days past the due date.

Previous Tasks (Already Implemented)

Write a program for a development company that sells properties. To do this, create a structure, named `PropertySale`, which records **UID** (unique identification number), **address**, **ZIP code**, **size**, **construction year**, and **price** of a flat that was sold. Create a database of `PropertySale` structures to store information, called `SalesDatabase`. The following operations can be performed on this database:

- Insert new flat sale information using a function, named Sales()
- Delete an entry in the database based on UID using a function, named Erase()
- Find an entry in the database using a function, called Search()
- Print an/all element(s) from the databse using a function, called PrintDB()
- GetZIP() and GetPrice() functions will allow access to the ZIP code and sales price of a flat
- Count the total sales in the database using a function called SalesCount()
- Compute the average prices for all the sales using a function, named AveragePrice()

New Tasks (Sorting)

1. **Sort** the properties in **ascending order** according to their price using a function named `SortByPriceAsc()`.
2. **Sort** the properties in **descending order** by price using a function called `SortByPriceDesc()`.
3. **Sort** properties within a specific range (by UID, price, etc.) using a function called `SortRange()`.

Submission Guidance: Prepare a document with your Name, Student ID, and Section, including the functions SortByPriceAsc(), SortByPriceDesc(), SortRange(), and their outputs. Bring a printed hardcopy to class on October 7 (Monday).

Rubrics

- **60% Execution**: Partial credit will be granted depending on how many of your methods work correctly.
- **30% Design**: The design score is based on how easy it is to follow the logic of your code, how well you avoided repetitive code, and how easy it would be to modify the code if certain specifications change.
- **10% Style**: Style includes comments, indentation, and the choice of variable and method names.

Substantial progress must be made towards correct execution to earn points for design and style.