PIMPRI CHINCHWAD EDUCATION TRUST's.

PIMPRI CHINCHWAD COLLEGE OF ENGINEERING

(An Autonomous Institute)

Class : SY BTech **Acad. Yr. 2025-26** **Semester : I**

Name of the student:Varad Amol Pisale **PRN : 124B1B043**

Department: Computer Engineering **Division : A**

Course Name : Data Structures Laboratory **Code:BCE23PC02**

Completion Date : 06/08/2025

Assignment No.

Problem Statement:

3 Merge sort A banking app needs to display a user's transaction history sorted by transaction amount to quickly identify large deposits or withdrawals.
Write a program for above scenario.

Hint:
Given a list of transaction amounts (positive and negative),implement Quick Sort to sort transactions in ascending order of amount. The solution should efficiently handle thousands of transactions.

Source Code :

```
#include <bits/stdc++.h>
using namespace std;

// partition
int partition(vector<long double> &arr, int low, int high)
{
   int pivot = arr[high];
   int i = low - 1;

   for (int j = low; j < high; j++)
   {
      if (arr[j] <= pivot)
      {
         i++;
         swap(arr[i], arr[j]);
      }
   }
   swap(arr[i + 1], arr[high]);
```

```
    return i + 1;
}

// quick sort
void quickSort(vector<long double> &arr, int low, int high)
{
    if (low < high)
    {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main()
{
    vector<long double> t; // long double to handle large transactions
    int n;

    cout << "Enter number of transactions: ";
    cin >> n;

    cout << "Enter transaction amounts :\n";
    for (int i = 0; i < n; i++)
    {
        long double amount;
        cin >> amount;
        t.push_back(amount);
    }

    quickSort(t, 0, n - 1);

    cout << "\nSorted transactions :\n";
    for (int i = 0; i < t.size(); i++)
    {
        cout << t[i] << " ";
    }
    cout << endl;

    return 0;
}
```

Screen Shot of Output :

```
Enter number of transactions: 3
Enter transaction amounts :
10000
50000
2500

Sorted transactions :
2500 10000 50000
```

Conclusion: Hence we have implemented a banking app needs to display a user's transaction history sorted by transaction amount