Name: Patil Krushna Gopal

Roll no.: 38

//Write a program for sorting given array in ascending/descending order using quick sort.

#include<iostream>

using namespace std;

class MergeSortDemo

{

    int A[16];

    int n;

public:

    void getData()

    {

        cout << "Enter the number of elements:";

        cin >> n;

        cout << "Enter the element:";

        for (int i = 0; i < n; i++)

        {

            cin >> A[i];

        }

    }

    void display()

    {

        cout << "Sorted elements in ascending order :";

        for (int i = 0; i < n; i++)

        {

            cout << A[i] << "\t";

        }

        cout << endl;

        cout << "Sorted elements in descending order :";

        for (int i = n; i >= 0; i--)

        {

            cout << A[i] << "\t";

        }

        cout << endl;

    }

    void merge(int A[], int temp[] ,int left, int mid, int right);

    void m\_sort(int A[], int temp[], int left, int right);

    void mergeSort();

};

void MergeSortDemo::merge(int A[], int temp[], int left, int mid, int right)

{

    int t\_pos, left\_end, n, i;

    t\_pos = left;

    left\_end = mid - 1;

    n = right - left + 1;

    while (left <= left\_end && mid <= right)

    {

        if (A[left] < A[mid])

        {

            temp[t\_pos] = A[left];

            t\_pos = t\_pos + 1;

            left = left + 1;

        }

        else

        {

            temp[t\_pos] = A[mid];

            t\_pos = t\_pos + 1;

            mid = mid + 1;

        }

    }

    while (left <= left\_end)

    {

        temp[t\_pos] = A[left];

        t\_pos = t\_pos + 1;

        left = left + 1;

    }

    while (mid <= right)

    {

        temp[t\_pos] = A[mid];

        t\_pos = t\_pos + 1;

        mid = mid + 1;

    }

    for (i = 0; i < n; i++)

    {

        A[right] = temp[right];

        right--;

    }

}

void MergeSortDemo::m\_sort(int A[], int temp[], int left, int right)

{

    int mid;

    if (right > left)

    {

        mid = (left + right) / 2;

        m\_sort(A, temp, left, mid);

        m\_sort(A, temp, mid + 1, right);

        merge(A, temp, left, mid + 1, right);

    }

}

void MergeSortDemo::mergeSort()

{

    int temp[10];

    m\_sort(A, temp, 0, n - 1);

}

int main(int argc, char\* argv[])

{

    MergeSortDemo o;

    o.getData();

    o.mergeSort();

    o.display();

}

Outuput:

Enter the number of elements:5

Enter the element:23 45 76 99 45

Sorted elements in ascending order :23 45 45 76 99

Sorted elements in descending order : 99 76 45 45 23