#include <iostream>

using namespace std;

template <typename T>

class Sorting

{

    T \*arr;

    int size;

public:

    Sorting(int n)

    {

        size = n;

        arr = new T[n];

    }

    void display()

    {

        for (int i = 0; i < size; i++)

            cout << arr[i] << " ";

        cout << endl;

    }

    void input()

    {

        for (int i = 0; i < size; i++)

            cin >> arr[i];

    }

    void selection\_sort()

    {

        int min\_idx;

        for (int i = 0; i < size; i++)

        {

            min\_idx = i;

            for (int j = i; j < size; j++)

            {

                if (arr[j] < arr[min\_idx])

                    min\_idx = j;

            }

            swap(arr[min\_idx], arr[i]);

        }

    }

    void bubble\_sort()

    {

        for (int i = 0; i < size; i++)

        {

            for (int j = 0; j < size - i - 1; j++)

            {

                if (arr[j] > arr[j + 1])

                    swap(arr[j], arr[j + 1]);

            }

        }

    }

    void insertion\_sort()

    {

        int temp, j;

        for (int i = 1; i < size; i++)

        {

            temp = arr[i];

            for (j = i - 1; j >= 0 && arr[j] > temp; j--)

                arr[j + 1] = arr[j];

            arr[j + 1] = temp;

        }

    }

};

int main()

{

    int n, choice;

    cout << "Enter size of array : ";

    cin >> n;

    Sorting<int> s(n);

    cout << "Enter elements of array" << endl;

    s.input();

    cout << "-----Select a sorting method-----" << endl;

    cout << "1. Insertion sort" << endl;

    cout << "2. Selection sort" << endl;

    cout << "3. Bubble sort" << endl;

    cin >> choice;

    switch (choice)

    {

    case 1:

        s.insertion\_sort();

        break;

    case 2:

        s.selection\_sort();

        break;

    case 3:

        s.bubble\_sort();

        break;

    default:

        cout << "Invalid choice Exiting program" << endl;

        break;

    }

    if (choice <= 3 && choice >= 1)

    {

        cout << "Sorted Array : ";

        s.display();

    }

    return 0;

}

