LP-II Mock Practical - 2

Name: Rohit Vilas Patil
Roll No. : 31353
Batch: M3 (TE3)
Title : Greedy Search Algorithm for Selection Sort Algorithm

Problem Statement: Implement Greedy Search Algorithm for any of the following Application :
1) Selection Sort

Code :

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

void displayArray(vector<int> &arr){
   int n = arr.size();
   for(int i=0;i<n;i++){
       cout << arr[i] << " ";
   }
   cout << endl;
}

void swap(vector<int> &arr, int i, int j){
   int temp = arr[i];
   arr[i] = arr[j];
   arr[j] = temp;
}

void selectionSort(vector<int> &arr){
   int n = arr.size();
   for(int i=0;i<n;i++){
       int minElement = arr[i];
       int min_idx = i;
       for(int j=i+1;j<n;j++){
           if(arr[j] < minElement){
               minElement = arr[j];
               min_idx = j;
           }
       }
       if(min_idx != i){
           swap(arr, i, min_idx);
       }
```

```
        cout << i+1 << "th Iteration: ";
        displayArray(arr);
    }
}

int main () {
    cout << "Enter the no. of elements in the array: ";
    int n;
    cin >> n;

    vector<int> arr(n);

    for(int i=0;i<n;i++){
        cout << i << " : ";
        cin >> arr[i];

    }
    cout << "Input Array : ";
    displayArray(arr);

    selectionSort(arr);

    cout << "Sorted Array : ";

    displayArray(arr);

    return 0;
}
```

Output:

pict@pict-OptiPlex-9020:~/Desktop/31353_Rohit_LP2_Mock_Practical$ cd "/home/pict/Desktop/31353_Rohit_LP2_Mock_Practical/" && g++ selectionSort.cpp -o selectionSort && "/home/pict/Desktop/31353_Rohit_LP2_Mock_Practical/"selectionSort
Enter the no. of elements in the array: 10
0 : 9
1 : 5
2 : 6
3 : 3
4 : 2

5 : 8
6 : 7
7 : 1
8 : 4
9 : 10
Input Array : 9 5 6 3 2 8 7 1 4 10
1th Iteration: 1 5 6 3 2 8 7 9 4 10
2th Iteration: 1 2 6 3 5 8 7 9 4 10
3th Iteration: 1 2 3 6 5 8 7 9 4 10
4th Iteration: 1 2 3 4 5 8 7 9 6 10
5th Iteration: 1 2 3 4 5 8 7 9 6 10
6th Iteration: 1 2 3 4 5 6 7 9 8 10
7th Iteration: 1 2 3 4 5 6 7 9 8 10
8th Iteration: 1 2 3 4 5 6 7 8 9 10
9th Iteration: 1 2 3 4 5 6 7 8 9 10
10th Iteration: 1 2 3 4 5 6 7 8 9 10
Sorted Array : 1 2 3 4 5 6 7 8 9 10