

Question4.cpp

```
1  //<----Lab 02- Sorting Techniques---->
2
3  // Q4. Create a single class Sort, which will provide the user the option to choose between
4  // all 4
5  // sorting techniques. The class should have following capabilities:
6  // * Take an array and a string (indicating the user choice for sorting technique) as
7  // input and perform the desired sorting.
8  // * Should allow the user to perform analysis on a randomly generated array. The
9  // analysis provides number of comparisons and number of swaps performed for
10 // each technique.
11 // * After printing all the results, the class should highlight the best and worst
12 // techniques.
13
14 #include<iostream>
15 using namespace std;
16 class Sort{
17     int* arr;
18     int n;
19     string choice;
20     int cmp[3],swap[3];
21 public:
22     Sort(int x): n(x){
23         arr = new int[x];
24     }
25     void bubbleSort(){
26         for(int i=0;i<n;i++){
27             cmp[0]++;
28             for(int j=0;j<n-i-1;j++){
29                 cmp[0]++;
30                 if(arr[j]>arr[j+1]){
31                     int temp;
32                     temp=arr[j];
33                     arr[j]=arr[j+1];
34                     arr[j+1]=temp;
35                     swap[0]++;
36                 }
37             }
38             cmp[0]++;
39         }
40     }
41     void InsertionSort(){
42         int key;
43         for(int i=1;i<n;i++){
44             cmp[1]++;
45             key=arr[i];
46             int j=i-1;
47             while( j >=0 && arr[j]>key){
48                 arr[j+1]=arr[j];
49                 j--;
50             }
51             cmp[1]++;
52             swap[1]++;
53             arr[j+1]=key;
54         }
55     }
56 }
```

```
53         swap[1]++;
54     }
55 }
56 void SelectionSort(){
57     int min_index=0;
58     for(int i=0;i<n;i++){
59         cmp[2]++;
60         min_index=i;
61         for(int j=i+1;j<n;j++){
62             cmp[2]++;
63             if(arr[j]<arr[min_index]){
64                 min_index=j;
65             }
66             cmp[2]++;
67         }
68         if(min_index!=i){
69             int temp;
70             temp=arr[i];
71             arr[i]=arr[min_index];
72             arr[min_index]=temp;
73             swap[2]++;
74         }
75     }
76 }
77 void choose(){
78     cout<<"Enter Data of Array:"<<endl;
79     for(int i=0;i<n;i++){
80         cin>>arr[i];
81     }
82     cout<<"\nBefore Sort:\n";
83     Display();
84     cout<<"enter choice b:bubble i:insertion s:selection"<<endl;
85     cin>>choice;
86     if(choice=="b"){
87         bubbleSort();
88     }
89     else if(choice=="i"){
90         InsertionSort();
91     }
92     else{
93         SelectionSort();
94     }
95     cout<<"\nAfter Sort:\n";
96     Display();
97 }
98 }
99 void comparison(){
100     for(int i=0;i<3;i++){
101         cmp[i]=0;
102         swap[i]=0;
103     }
104     int temp[n];
105     for(int i=0;i<n;i++){
106         arr[i]=rand();
107         temp[i]=arr[i];
108     }
```

```

109     bubbleSort();
110     for(int i=0;i<n;i++){
111         arr[i]=temp[i];
112     }
113     InsertionSort();
114     for(int i=0;i<n;i++){
115         arr[i]=temp[i];
116     }
117     SelectionSort();
118
119     int BestS=99999,WorstS=-1,BestC=99999,WorstC=-1;
120     int tBS,tWS,tBC,tWC;
121
122     for(int i=0;i<3;i++){
123         if(swap[i]<BestS){
124             BestS=swap[i];tBS=i;
125         }
126         if(swap[i]>WorstS){
127             WorstS=swap[i];tWS=i;
128         }
129         if(cmp[i]<BestC){
130             BestC=cmp[i];tBC=i;
131         }
132         if(cmp[i]>WorstC){
133             WorstC=cmp[i];tWC=i;
134         }
135     }
136
137     cout<<"Test Array:\n";
138     for(int i=0;i<n;i++){
139         cout<<temp[i]<<" ";
140     }
141     cout<<endl<<endl;
142
143     cout<<"Bubble Sort for "<<n<<" Size array - Comparisons: "<<cmp[0]<<" Swaps: "<<
swap[0]<<endl;
144     cout<<"Insertion Sort for "<<n<<" Size array - Comparisons: "<<cmp[1]<<" Swaps:
"<<swap[1]<<endl;
145     cout<<"Selection Sort for "<<n<<" Size array - Comparisons: "<<cmp[2]<<" Swaps:
"<<swap[2]<<endl<<endl<<endl;
146     cout<<"Technique 0 = BubbleSort, 1 = InsertionSort, 2 = SelectionSort"<<endl<<
endl;
147     cout<<"Best Technique (Comparisons) : "<<tBC<<" - at "<<BestC<<" Comparisons."<<
endl<<endl;
148     cout<<"Worst Technique (Comparisons) : "<<tWC<<" - at "<<WorstC<<" Comparisons."<
<endl<<endl;
149     cout<<"Best Technique (Swaps) : "<<tBS<<" - at "<<BestS<<" Swaps."<<endl<<endl;
150     cout<<"Worst Technique (Swaps) : "<<tWS<<" - at "<<WorstS<<" Swaps."<<endl<<endl;
151
152     }
153
154     void Display(){
155         for(int i=0;i<n;i++){
156             cout<<arr[i]<<" ";
157         }
158         cout<<endl<<endl;
159     }

```

```
160 };
161
162 int main(){
163     int n;
164     cout<<"Enter Size of Array to sort:\n";
165     cin>>n;
166     Sort x(n);
167     x.choose();
168     cout<<"\n\nEnter Size of Array to Compare sorts:\n";
169     cin>>n;
170     Sort y(n);
171     y.comparison();
172 }
```