

	Department of Computer Science and Engineering Faculty of Engineering, South Eastern University of Sri Lanka			
	Subject	CS53003: Data Structure and Algorithms		
	Batch	E18	Semester	5

Lab no and title : Lab 03: Sorting
Name : G.W.P.R.R. Wijesinghe
Reg No : SEU-IS-18-EG-013
Submission Date : 10-01-2022

-
1. Illustrate the execution of the insertion-sort algorithm on the array $A = \{3, 13, 89, 34, 44, 99, 9\}$, writing the intermediate values of A at each iteration of the algorithm.

Original Array	3	13	89	34	44	99	9
Pass 1	3	13	89	34	44	99	9
Pass 2	3	13	89	34	44	99	9
Pass 3	3	13	34	89	44	99	9
Pass 4	3	13	34	44	89	99	9
Pass 5	3	13	34	44	89	99	9
Pass 6	3	9	13	34	44	89	99

2. Use Bubble sort and Selection sort to sort the following list of numbers into ascending order by writing the intermediate values:

3 6 2 7 4 1 5

Sorting method: _____Bubble sort_____

Original Array	3	6	2	7	4	1	5
Pass 1	3	2	6	4	1	5	7
Pass 2	2	3	4	1	5	6	7
Pass 3	2	3	1	4	5	6	7
Pass 4	2	1	3	4	5	6	7
Pass 5	1	2	3	4	5	6	7
Pass 6	1	2	3	4	5	6	7

Sorting method: _____Selection sort_____

Original Array	3	6	2	7	4	1	5
Pass 1	1	6	2	7	4	3	5
Pass 2	1	2	6	7	4	3	5
Pass 3	1	2	3	7	4	6	5
Pass 4	1	2	3	4	7	6	5
Pass 5	1	2	3	4	5	6	7
Pass 6	1	2	3	4	5	6	7

3. Implement the following sorting Algorithms in C++. (Note: Array of inputs should be given on run time)
- Bubble sort

```
Start here X BubbleSorting.cpp X SelectionSort.cpp X InsertionSorting.cpp X
1  #include<iostream>
2
3  using namespace std;
4
5  int printArray(int Arr[], int size){
6      int i;
7      for (i = 0; i < size; i++){
8          cout << Arr[i] << " ";
9      }
10 }
11
12 int BubbleSort(int Arr[],int n){
13     for (int k=0; k<n-1; k++){
14         for (int i=0; i<n-k-1; i++){
15             if (Arr[i] > Arr[i+1]){
16                 swap(Arr[i],Arr[i+1]);
17             }
18         }
19     }
20 }
21
22 int main(){
23     int m;
24     cout << "How many numbers do you want to sort : ";
25     cin >> m;
26     int B[m] ={};
27     for (int i=0; i<m; i++){
28         cout << "Enter your number : ";
29         cin >> B[i];
30     }
31     cout << "Unsorted array : ";
32     printArray(B,m);
33     BubbleSort(B,m);
34     cout << "\nSorted array :";
35     printArray(B,m);
36 }
37
```

```
"E:\Campus Semesters\5th Semester\CS 53003 Data Structure and Algorithms\Lab\03\BubbleSorting.exe"
How many numbers do you want to sort : 8
Enter your number : 96
Enter your number : 45
Enter your number : 28
Enter your number : 74
Enter your number : 20
Enter your number : 16
Enter your number : 11
Enter your number : 85
Unsorted array : 96 45 28 74 20 16 11 85
Sorted array :11 16 20 28 45 74 85 96
Process returned 0 (0x0)   execution time : 34.273 s
Press any key to continue.
```

- Selection sort

```

Start here X BubbleSorting.cpp X SelectionSort.cpp X InsertionSorting.cpp X
1  #include<iostream>
2
3  using namespace std;
4
5  int printArray(int Arr[], int size){
6      int i;
7      for (i = 0; i < size; i++){
8          cout << Arr[i] << " ";
9      }
10 }
11 int SelectionSort(int Arr[], int n){
12     for (int i=0; i<n-1; i++){
13         int Minimum = i;
14         for (int j=i+1; j<n; j++){
15             if (Arr[j] < Arr[Minimum]){
16                 Minimum = j;
17             }
18         }
19         if (Minimum != i){
20             swap(Arr[i],Arr[Minimum]);
21         }
22     }
23 }
24
25 int main(){
26     int m;
27     cout << "How many numbers do you want to sort : ";
28     cin >> m;
29     int B[m] ={};
30     for (int i=0; i<m; i++){
31         cout << "Enter your number : ";
32         cin >> B[i];
33     }
34     cout << "Unsorted array : ";
35     printArray(B,m);
36     SelectionSort(B,m);
37     cout << "\nSorted array :";
38     printArray(B,m);
39 }
40

```

```

"E:\Campus Semseters\5th Semester\CS 53003 Data Structure and Algorithms\Lab\03\SelectionSort.exe"
How many numbers do you want to sort : 8
Enter your number : 56
Enter your number : 32
Enter your number : 47
Enter your number : 99
Enter your number : 11
Enter your number : 86
Enter your number : 51
Enter your number : 14
Unsorted array : 56 32 47 99 11 86 51 14
Sorted array :11 14 32 47 51 56 86 99
Process returned 0 (0x0)   execution time : 19.246 s
Press any key to continue.

```

- Insertion sort

```

Start here X BubbleSorting.cpp X SelectionSort.cpp X InsertionSorting.cpp X
1  #include<iostream>
2
3  using namespace std;
4
5  int printArray(int Arr[], int size){
6      int i;
7      for (i = 0; i < size; i++){
8          cout << Arr[i] << " ";
9      }
10 }
11
12 int InsectionSort(int Arr[],int n){
13     for (int i=1; i<n; i++){
14         int key = Arr[i];
15         int j = i;
16         while (j>0 && Arr[j-1] > key){
17             Arr[j] = Arr [j-1];
18             j--;
19         }
20         Arr[j] = key;
21     }
22 }
23
24 int main(){
25     int m;
26     cout << "How many numbers do you want to sort : ";
27     cin >> m;
28     int B[m] ={};
29     for (int i=0; i<m; i++){
30         cout << "Enter your number : ";
31         cin >> B[i];
32     }
33     cout << "Unsorted array : ";
34     printArray(B,m);
35     InsectionSort(B,m);
36     cout << "\nSorted array :";
37     printArray(B,m);
38 }
39

```

```

E:\Campus Semseters\5th Semester\CS 53003 Data Structure and Algorithms\Lab\03\InsertionSorting.exe
How many numbers do you want to sort : 8
Enter your number : 87
Enter your number : 64
Enter your number : 25
Enter your number : 98
Enter your number : 31
Enter your number : 25
Enter your number : 11
Enter your number : 63
Unsorted array : 87 64 25 98 31 25 11 63
Sorted array :11 25 25 31 63 64 87 98
Process returned 0 (0x0)   execution time : 17.485 s
Press any key to continue.

```

4. Rewrite the Insertion-Sort algorithm to sort into decreasing order.

```
Start here X BubbleSorting.cpp X SelectionSort.cpp X InsertionSortingDis.cpp X
1  #include<iostream>
2
3  using namespace std;
4
5  int printArray(int Arr[], int size){
6      int i;
7      for (i = 0; i < size; i++){
8          cout << Arr[i] << " ";
9      }
10 }
11
12 int InsectionSort(int Arr[],int n){
13     for (int i=1; i<n; i++){
14         int key = Arr[i];
15         int j = i;
16         while (j>0 && Arr[j-1] < key){
17             Arr[j] = Arr [j-1];
18             j--;
19         }
20         Arr[j] = key;
21     }
22 }
23
24 int main(){
25     int m;
26     cout << "How many numbers do you want to sort : ";
27     cin >> m;
28     int B[m] ={};
29     for (int i=0; i<m; i++){
30         cout << "Enter your number : ";
31         cin >> B[i];
32     }
33     cout << "Unsorted array : ";
34     printArray(B,m);
35     InsectionSort(B,m);
36     cout << "\nSorrted array :";
37     printArray(B,m);
38 }
39
```

```
"E:\Campus Semseters\5th Semester\CS 53003 Data Structure and Algorithms\Lab\03\InsertionSortingDis.exe"
How many numbers do you want to sort : 8
Enter your number : 15
Enter your number : 95
Enter your number : 74
Enter your number : 36
Enter your number : 55
Enter your number : 28
Enter your number : 11
Enter your number : 66
Unsorted array : 15 95 74 36 55 28 11 66
Sorrted array :95 74 66 55 36 28 15 11
Process returned 0 (0x0)   execution time : 225.131 s
Press any key to continue.
```
