LAB TASK # 04

MOHAMMAD BASIL ALI KHAN

20K-0477

Task#01:

- **Main Menu**

```
92
93  int main()
94  {
95      int List[]={5, 3, 8, 1, 4, 6, 2, 7, 10, 9};
96      int size, choice, temp;
97      size = sizeof(List)/sizeof(int);
98      cout << "***** SORTING ******" << endl << endl;
99      cout << "1 -> Bubble Sort. " << endl;
100     cout << "2 -> Insertion Sort. " << endl;
101     cout << "3 -> Selection Sort. " << endl;
102     cout << "4 -> Comb Sort. " << endl << endl;
103     cout << "List: " << endl;
104     Display(List, size);
105     cout << endl << endl << "Which sorting to use to sort above List ?: ";
106     cin >> choice;
107     switch(choice)
108     {
109         case 1:
110             {
111                 BubbleSort(List, size);
112                 Display(List, size);
113                 break;
114             }
115         case 2:
116             {
117                 InsertionSort(List, size);
118                 Display(List, size);
119                 break;
120             }
```

```
112                 Display(List, size);
113                 break;
114             }
115         case 2:
116             {
117                 InsertionSort(List, size);
118                 Display(List, size);
119                 break;
120             }
121         case 3:
122             {
123                 SelectionSort(List, size);
124                 Display(List, size);
125                 break;
126             }
127         case 4:
128             {
129                 CombSort(List, size);
130                 Display(List, size);
131                 break;
132             }
133         default:
134             {
135                 cout << endl << "Invalid Choice for sorting!!! " << endl;
136                 break;
137             }
138     }
139 }
```

```
E:\FAST\3rd Semester\DS Lab\Lab Task 04\Task#01.exe
***** SORTING ******

1 -> Bubble Sort.
2 -> Insertion Sort.
3 -> Selection Sort.
4 -> Comb Sort.

List:
5 3 8 1 4 6 2 7 10 9

Which sorting to use to sort above List ?:
```

- **Bubble Sort**

```
12
13  void BubbleSort(int arr[], int size)
14  {
15      system("cls");
16      cout << endl << "Bubble Sort: " << endl << endl;
17      for(int i=0; i<size-1; i++)
18          {
19              for(int j=i+1; j<size; j++)
20                  {
21                      if(arr[i]>arr[j])
22                      {
23                          int temp = arr[i];
24                          arr[i] = arr[j];
25                          arr[j] = temp;
26                      }
27                  }
28          }
29  }
30
```

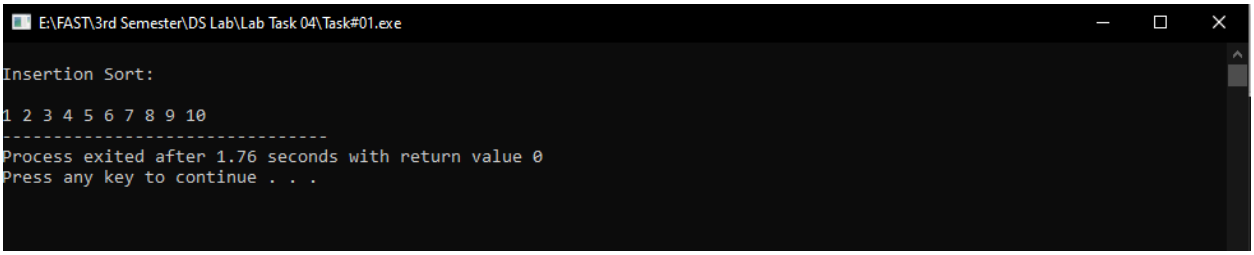
```
E:\FAST\3rd Semester\DS Lab\Lab Task 04\Task#01.exe

Bubble Sort:

1 2 3 4 5 6 7 8 9 10
--------------------------------
Process exited after 47.38 seconds with return value 0
Press any key to continue . . .
```

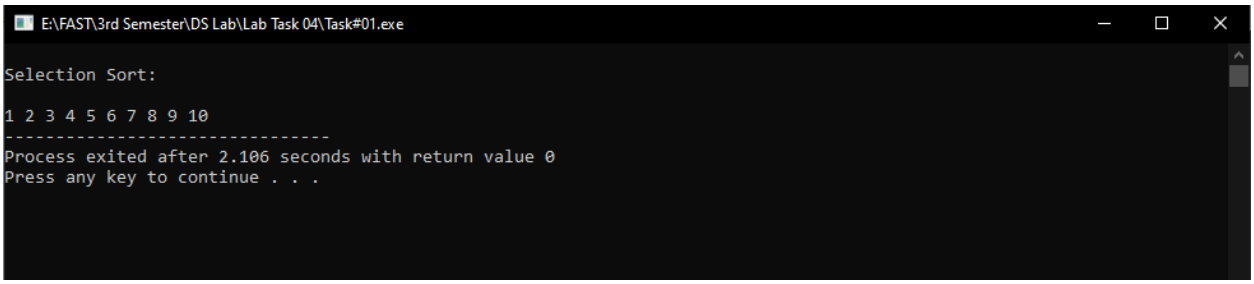
- **Insertion Sort**

```
30
31  void InsertionSort(int arr[], int size)
32  {
33      system("cls");
34      cout << endl << "Insertion Sort: " << endl << endl;
35      for(int i=1; i<size; i++)
36          {
37              int temp = arr[i];
38              int j;
39              for(j=i-1; j>=0 && arr[j]>temp; j--)
40              {
41                  arr[j+1]= arr[j];
42              }
43              arr[j+1] = temp;
44          }
45  }
```



- **Selection Sort**

```
void SelectionSort(int arr[], int size)
{
    system("cls");
    cout << endl << "Selection Sort: " << endl << endl;
    for(int i=0; i<size-1; i++)
    {
        int min=i;
        for(int j=i+1; j<size; j++)
        {
            if(arr[j]<arr[min])
            {
                min = j;
            }
        }
        if(min != i)
        {
            int temp = arr[i];
            arr[i] = arr[min];
            arr[min] = temp;
        }
    }
}
```



- **Comb Sort**

```
void CombSort(int arr[], int size)
{
    system("cls");
    cout << endl << "Comb Sort: " << endl << endl;
    int gap = size;
    bool swapped = false;
    const float SHRINK_FACTOR = 1.3;
    while(gap > 1 || swapped == true)
    {
        swapped = false;
        gap = int(gap / SHRINK_FACTOR);
        for(int i=0; i<(size-gap); i++)
        {
            if(arr[i]>arr[i+gap])
            {
                swap(arr[i],arr[i+gap]);
                swapped = true;
            }
        }

    }
}
```

```
E:\FAST\3rd Semester\DS Lab\Lab Task 04\Task#01.exe

Comb Sort:

1 2 3 4 5 6 7 8 9 10
--------------------------------
Process exited after 1.988 seconds with return value 0
Press any key to continue . . .
```

Task#02:

- **Code**

Task#02.cpp

```cpp
#include<iostream>

using namespace std;

int Partition(int arr[], int low, int high)
{
    int pivot = arr[high];
    int i = (low - 1);

    for (int j = low; j <= high- 1; j++)
    {
        if (arr[j] <= pivot)
        {
            i++;
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return (i + 1);
}

void QuickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pivot = Partition(arr, low, high);
        QuickSort(arr, low, pivot - 1);
        QuickSort(arr, pivot + 1, high);
    }
```

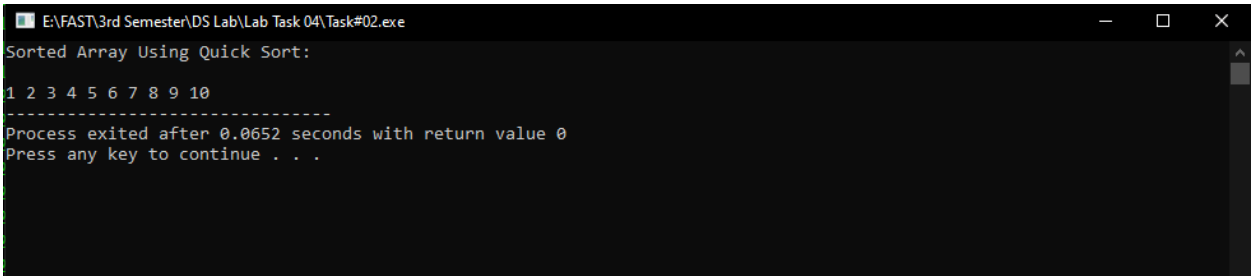
Task#02.cpp

```cpp
            swap(arr[i], arr[j]);
        }
    }
    swap(arr[i + 1], arr[high]);
    return (i + 1);
}

void QuickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pivot = Partition(arr, low, high);
        QuickSort(arr, low, pivot - 1);
        QuickSort(arr, pivot + 1, high);
    }
}

int main()
{
    int List[]={5, 3, 8, 1, 4, 6, 2, 7, 10, 9};
    int size = sizeof(List)/sizeof(int);
    QuickSort(List, 0, size-1);
    cout << "Sorted Array Using Quick Sort: " << endl << endl;
    for (int i=0; i < size; i++)
    {
        cout << List[i] << " ";
    }
}
```

- **Output**


E:\FAST\3rd Semester\DS Lab\Lab Task 04\Task#02.exe
Sorted Array Using Quick Sort:
1 2 3 4 5 6 7 8 9 10
Process exited after 0.0652 seconds with return value 0
Press any key to continue . . .