NAME - AKRITI CHOUDHARY

ROLL NUMBER - 2005776

SUBJECT - DSA LAB

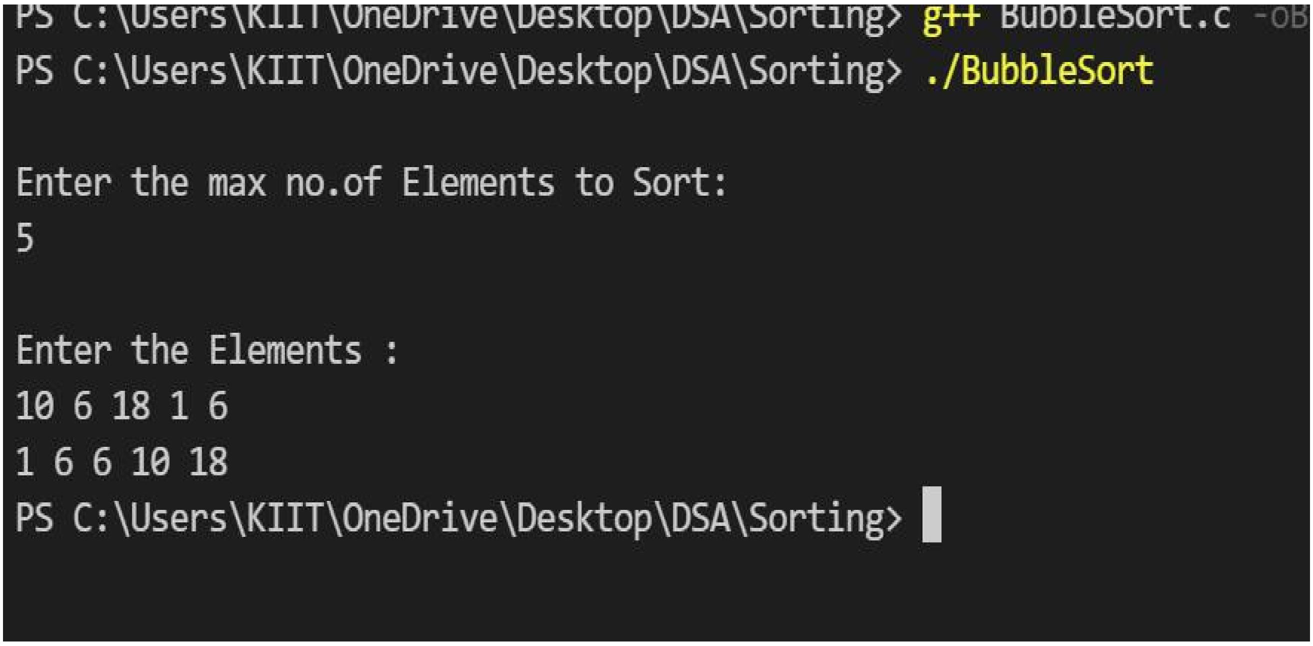
DATE - 16/11/2021

CLASS - B14

BRANCH - CSE

Bubble Sort

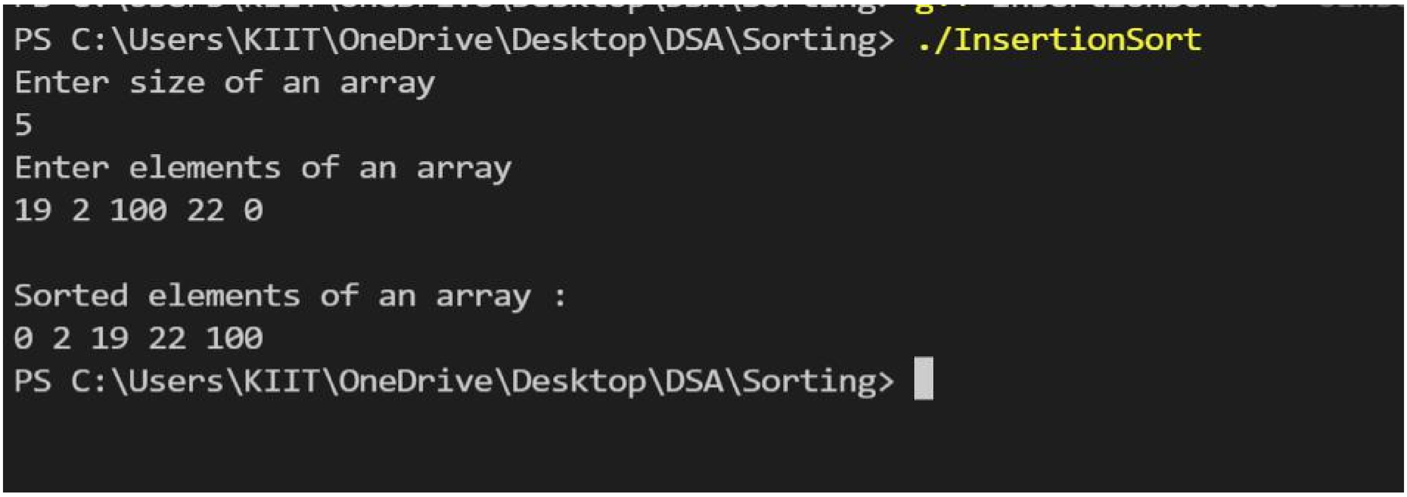
```
#include <stdio.h>

int main()
{
   int i, j, temp, n;
   printf("\nEnter the max no.of Elements to Sort: \n");
   scanf("%d", &n);
   int a[n];

   puts("\nEnter the Elements : ");
   for (i = 0; i < n; i++)
   {
      scanf("%d", &a[i]);
   }
   for (i = 0; i < n; i++)
      for (j = i + 1; j < n; j++)
      {
         if (a[i] > a[j])
         {
            temp = a[i];
            a[i] = a[j];
            a[j] = temp;
         }
      }
   for (i = 0; i < n; i++)
   {
      printf("%d ", a[i]);
   }
   return 0;
}
```



```
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> g++ BubbleSort.c -oB
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> ./BubbleSort

Enter the max no.of Elements to Sort:
5

Enter the Elements :
10 6 18 1 6
1 6 6 10 18
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting>
```

Insertion Sort

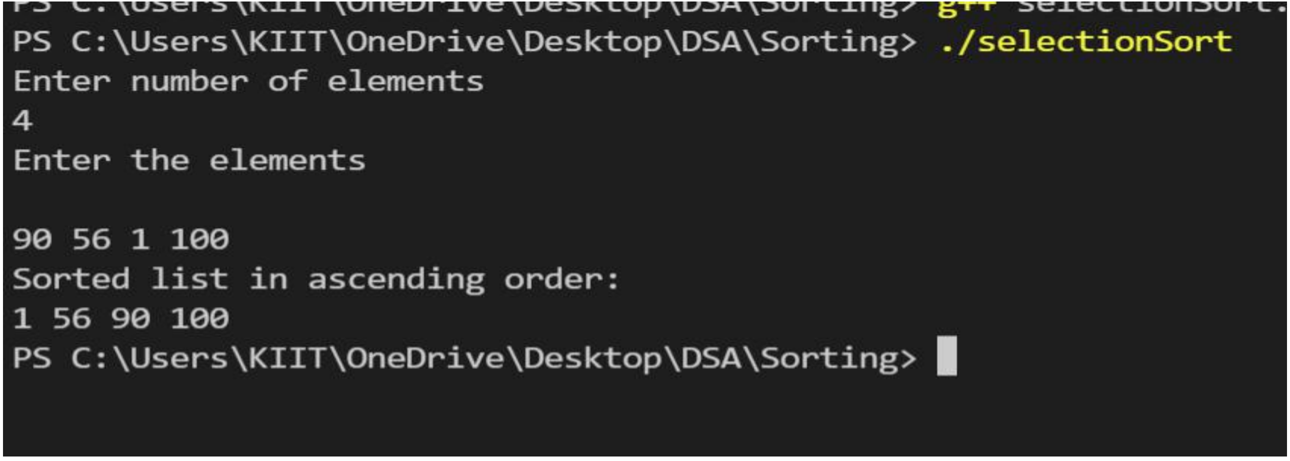
```
#include <stdio.h>

int main()
{
   int n;
   int key, temp, j;
   puts("Enter size of an array");
   scanf("%d", &n);
   int arr[n];
   puts("Enter elements of an array");
   for (int i = 0; i < n; ++i)
   {
      scanf("%d", &arr[i]);
   }
   puts("");
   //sorting
   for (int i = 0; i < n; ++i)
   {
      key = arr[i];
      j = i - 1;
      while ((j >= 0) && (arr[j] > key))
      {
         arr[j + 1] = arr[j];
         j--;
         arr[j + 1] = key;
      }
   }
   //printing
   puts("Sorted elements of an array :");
   for (int i = 0; i < n; ++i)
   {
      printf("%d ", arr[i]);
   }
   return 0;
}
```



```
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> ./InsertionSort
Enter size of an array
5
Enter elements of an array
19 2 100 22 0

Sorted elements of an array :
0 2 19 22 100
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting>
```

Selection Sort

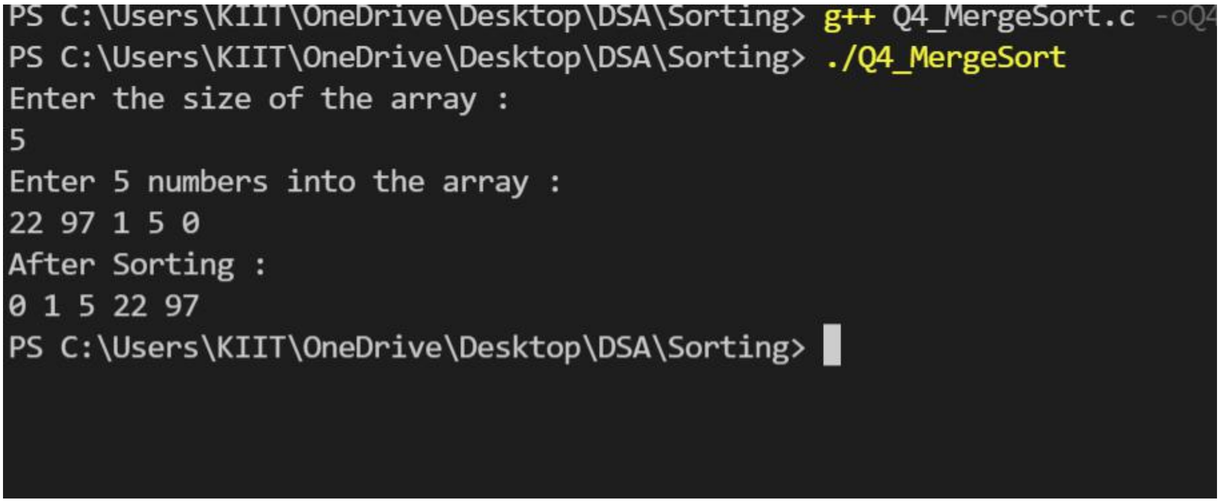
```
#include <stdio.h>
int main()
{
   int n, i, j, position, t;
   puts("Enter number of elements");
   scanf("%d", &n);
   int array[n];
   puts("Enter the elements\n");

   for (i = 0; i < n; i++)
      scanf("%d", &array[i]);

   for (i = 0; i < (n - 1); i++)
   {
      position = i;

      for (j = i + 1; j < n; j++)
      {
         if (array[position] > array[j])
            position = j;
      }
      if (position != i)
      {
         t = array[i];
         array[i] = array[position];
         array[position] = t;
      }
   }
   puts("Sorted list in ascending order:");
   for (i = 0; i < n; i++)
      printf("%d ", array[i]);

   puts(" ");

   return 0;
}
```



```
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> g++ selectionSort.
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> ./selectionSort
Enter number of elements
4
Enter the elements

90 56 1 100
Sorted list in ascending order:
1 56 90 100
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting>
```

Merge Sort

```
#include <stdio.h>
#include <stdlib.h>

int n;
int *arr;
int *brr;

void input()
{
    int i;

    printf("Enter %d numbers into the array : \n", n);

    for (i = 0; i < n; i++)
    {
        scanf("%d", &arr[i]);
    }
}

void display()
{
    int i;

    for (i = 0; i < n; i++)
    {
        printf("%d ", arr[i]);
    }

    printf("\n");
}

void merge(int low, int mid, int high)
{
    int l1, l2, i;

    for (l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++)
    {
        if (arr[l1] <= arr[l2])
            brr[i] = arr[l1++];

        else
            brr[i] = arr[l2++];
    }

    while (l1 <= mid)
    {
        brr[i++] = arr[l1++];
    }

    while (l2 <= high)
    {
        brr[i++] = arr[l2++];
    }

    for (i = low; i <= high; i++)
    {
        arr[i] = brr[i];
    }
}

void sort(int low, int high)
{
    int mid;

    if (low < high)
    {
        mid = (low + high) / 2;
        sort(low, mid);
        sort(mid + 1, high);
        merge(low, mid, high);
    }
}

int main()
{
    puts("Enter the size of the array :");
    scanf("%d", &n);

    arr = (int *)malloc(n * sizeof(int));
    brr = (int *)malloc(n * sizeof(int));

    input();

    sort(0, n - 1);

    puts("After Sorting : ");
    display();

    return 0;
}
```



```
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> g++ Q4_MergeSort.c -oQ4
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> ./Q4_MergeSort
Enter the size of the array :
5
Enter 5 numbers into the array :
22 97 1 5 0
After Sorting :
0 1 5 22 97
PS C:\Users\KIIT\OneDrive\Desktop\DSA\Sorting> 
```