ASSIGNMENT - 6

NAME: AKSHAT JAIMINI

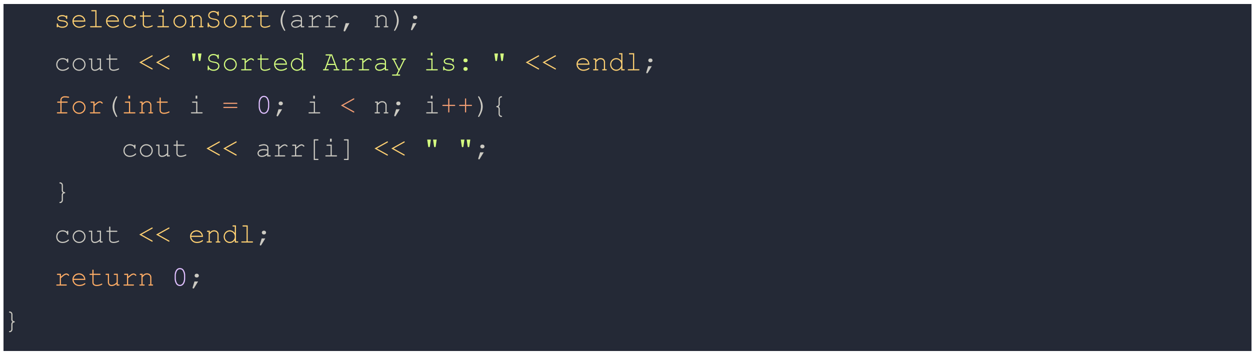
ROLL NUMBER: 102103586

Q1. Selection Sort

```cpp
#include <iostream>
using namespace std;

void selectionSort(int* arr, int len){
    for(int i = 0; i < len - 1; i++){
        int minInd = i;
        for(int j = i+1; j < len; j++){
            if(arr[minInd] > arr[j]){
                minInd = j;
            }
        }
        if(minInd != i){
            int temp = arr[minInd];
            arr[minInd] = arr[i];
            arr[i] = temp;
        }
    }
}

int main(){
    int n;
    cout << "Enter the number of elements" << endl;
    cin >> n;
    int arr[n];
    for(int i = 0; i < n; i++){
        cout << "[" << i << "]: ";
        cin >> arr[i];
    }
```



```
    selectionSort(arr, n);
    cout << "Sorted Array is: " << endl;
    for(int i = 0; i < n; i++){
        cout << arr[i] << " ";
    }
    cout << endl;
    return 0;
}
```

OUTPUT:

```
➜  lab6 git:(main) ✗ ./a.out
Enter the number of elements
10
[0]: 45
[1]: 32
[2]: 46
[3]: 3
[4]: 23
[5]: 52
[6]:
556
[7]: 5
[8]: 5
[9]: 5
Sorted Array is:
3 5 5 5 23 32 45 46 52 556
```

Q2. Insertion Sort

```cpp
#include <iostream>
using namespace std;

void insertionSort(int* arr, int n){
    for(int i = 1; i < n; i++){
        int curInd = i;
        for(int j = i; j >= 0; j--){
            if(arr[curInd] < arr[j]){
                int temp = arr[curInd];
                arr[curInd] = arr[j];
                arr[j] = temp;
                curInd = j;
            }
        }
    }
}
int main(){
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    int* arr = new int[n];
    cout << "Enter the values in the array" << endl;
    for(int i = 0; i < n; i++){
        cout << "[" << i << "]: ";
        cin >> arr[i];
    }
    insertionSort(arr,n);
    cout << "Sorted Array is: " << endl;
    for(int i = 0; i < n; i++){
        cout << arr[i];
        if(i == n-1){
            cout << endl;
        }else{
            cout << ", ";
        }
    }
    delete[] arr;
}
```

OUTPUT:

```
➜  lab6 git:(main) ✗ ./a.out
Enter the number of elements: 10
Enter the values in the array
[0]: 34
[1]: 35
[2]: 2
[3]: 34
[4]: 12
[5]: 3
[6]: 43
[7]: 6
[8]: 575
[9]: 56
Sorted Array is:
2, 3, 6, 12, 34, 34, 35, 43, 56, 575
```

Q3. Bubble Sort

```
#include <iostream>
using namespace std;


void bubbleSort(int* arr, int n){
    for(int i = 0; i < n - 1; i++){
        for(int j = 0; j< n; j++){
            if(arr[j] > arr[j+1]){
                int temp = arr[j];
                arr[j] = arr[j+1];
                arr[j+1] = temp;
            }
        }
    }
}
int main(){
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    int* arr = new int[n];
    for(int i = 0; i < n; i++){
        cin >> arr[i];
    }
    cout << "Sorted array: " << endl;
    bubbleSort(arr, n);
    for(int i = 0; i < n; i++){
        cout << arr[i];
        if(i == n-1)
            cout << endl;
        else
            cout << ", ";
    }
    delete[] arr;
    return 0;
}
```

OUTPUT

```
➜  lab6 git:(main) ✗ ./a.out
Enter the number of elements: 10
34
5
23
5
23
1
56
46
43
34
Sorted array:
1, 5, 5, 23, 23, 34, 34, 43, 46, 56
```

Q4. MERGE SORT

```
#include <iostream>
using namespace std;

void merge(int* arr, int l, int mid, int h){
    int arrLeft[mid-l+1];
    int arrRight[h - mid];
    int len1 = mid-l+1;
    int len2 = h - mid;
    for(int m = 0; m <= mid; m++){
        arrLeft[m] = arr[l+m];
    }
    for(int m = 0; m <= mid; m++){
        arrRight[m] = arr[mid+1+m];
    }
    int arrComb[h-l+1];
    int i = 0, j = 0, k = 0;
    while(i < len1 && j < len2){
        if(arrLeft[i] < arrRight[j]){
            arr[k] = arrLeft[i];
            k++;
            i++;
        }else{
            arr[k] = arrRight[j];
            j++;
            k++;
        }
    }
    while(i < len1){
        arr[k] = arr[i];
        k++;
        i++;
    }
    while(j < len2){
        arr[k] = arr[j];
        k++;
        j++;
    }
}

void mergeSort(int* arr, int l, int h){
   if(l >= h)
       return;
   int mid = (l+h)/2;
   mergeSort(arr, l,mid);
   mergeSort(arr, mid+1, h);
   merge(arr,l,mid,h);
}
int main(){
   int n;
   cout << "Enter the number of elements in the array" << endl;
   cin >> n;
   int arr[n];
   cout << "Copying" << endl;
   for(int i = 0; i < n; i++){
       cout << "[" << i << "]: ";
       cin >> arr[i];
   }
   cout << "Done" << endl;
   mergeSort(arr,0,n-1);
   for(int i = 0; i < n; i++){
       cout << arr[i];
       if(i == n-1)
           cout << endl;
       else
           cout << ", ";
   }
}
```

OUTPUT

```
➜  lab6 git:(main) ✗  ./a.out
Enter the number of elements: 10
34
5
23
5
23
1
56
```

```
46
43
34
Sorted array:
1, 5, 5, 23, 23, 34, 34, 43, 46, 56
```

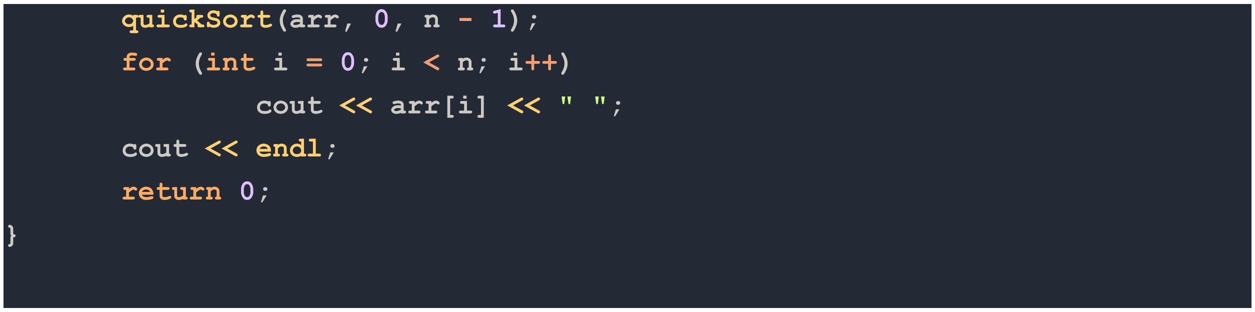
Q5. QuickSort

```
#include <iostream>
using namespace std;

void quickSort(int arr[], int left, int right) {
        int i = left, j = right;
        int tmp;
        int pivot = arr[(left + right) / 2];

        /* partition */
        while (i <= j) {
                while (arr[i] < pivot)
                        i++;
                while (arr[j] > pivot)
                        j--;
                if (i <= j) {
                        tmp = arr[i];
                        arr[i] = arr[j];
                        arr[j] = tmp;
                        i++;
                        j--;
                }
        };
        if (left < j)
                quickSort(arr, left, j);
        if (i < right)
                quickSort(arr, i, right);
}

int main() {
        int n;
        cout << "Enter the number of elements" << endl;
        cin >> n;
        int arr[n];
        cout << "Enter elements" << endl;
        for(int i = 0; i < n; i++){
            cout << "[" << i << "]: ";
            cin >> arr[i];
        }
```



```
	quickSort(arr, 0, n - 1);
	for (int i = 0; i < n; i++)
		cout << arr[i] << " ";
	cout << endl;
	return 0;
}
```

```
➜ lab6 git:(main) ✗ ./a.out
Enter the number of elements
10
Enter elements
[0]: 3
[1]: 46
[2]: 57
[3]: 34
[4]: 6
[5]: 452
[6]: 4
[7]: 5
[8]: 34556
[9]: 54
3 4 5 6 34 46 54 57 452 34556
```