Name:Eliza Arora

Group:2Q-21

Roll No-102317299

Assignment-1

Q1)

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

int main() {
   int arr[] = {2, 5, 8, 12, 16, 23, 38, 56, 72, 91};
   int n = sizeof(arr) / sizeof(arr[0]);
   int target = 23;

   int left = 0, right = n - 1, mid;
   while (left <= right) {
      mid = left + (right - left) / 2;
      if (arr[mid] == target) {
         cout << "Target found at index: " << mid << endl;
         return 0;
      } else if (arr[mid] < target) {
         left = mid + 1;
      } else {
         right = mid - 1;
      }
   }

   cout << "Target not found" << endl;
   return 0;
}
```

```
Target found at index: 5
```

Q2)

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

void merge(int arr[], int left, int mid, int right) {
    int n1 = mid - left + 1;
    int n2 = right - mid;
    int L[n1], R[n2];

    for (int i = 0; i < n1; i++) L[i] = arr[left + i];
    for (int i = 0; i < n2; i++) R[i] = arr[mid + 1 + i];

    int i = 0, j = 0, k = left;
    while (i < n1 && j < n2) {
        if (L[i] <= R[j]) arr[k++] = L[i++];
        else arr[k++] = R[j++];
    }

    while (i < n1) arr[k++] = L[i++];
    while (j < n2) arr[k++] = R[j++];
}

void mergeSort(int arr[], int left, int right) {
    if (left < right) {
        int mid = left + (right - left) / 2;
        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);
```

```
        merge(arr, left, mid, right);
    }
}

int main() {
    int arr[] = {12, 11, 13, 5, 6, 7};
    int n = sizeof(arr) / sizeof(arr[0]);

    mergeSort(arr, 0, n - 1);

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    return 0;
}
```

```
5 6 7 11 12 13
```

Q3)

```
#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; 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 pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

int main() {
    int arr[] = {4, 2, 6, 9, 2};
    int n = sizeof(arr) / sizeof(arr[0]);

    quickSort(arr, 0, n - 1);

    for (int i = 0; i < n; i++) cout << arr[i] << " ";
    return 0;
}
```

```
2 2 4 6 9
```

Q4)

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

```
int main() {
    int arr[] = {-2, -5, 6, -2, -3, 1, 5, -6};
    int n = sizeof(arr) / sizeof(arr[0]);

    int max_sum = INT_MIN, current_sum = 0;

    for (int i = 0; i < n; i++) {
        current_sum += arr[i];
        if (current_sum > max_sum) max_sum = current_sum;
        if (current_sum < 0) current_sum = 0;
    }

    cout << "Maximum subarray sum is: " << max_sum << endl;

    return 0;
}
```

```
7
```