Assignment - 1

Name - Samar Goyal
Roll no - 1024030992

Ans 1: algo of binary search

```
int binary_search(int arr[],int low,int high,int key){
    if(low>high){
        return -1;
    }
    int mid = (low+high)/2;

    if(arr[mid]==key){
        return mid;
    }
    else if(arr[mid]<key){
      return binary_search(arr,mid+1,high,key);
    }
    else{
       return binary_search(arr,low,mid-1,key);
    }

}
```

Output:

```
Enter Size of Array: 5
1 4 6 3 2
Enter key:4
Key present at index: 1%
```

Assignment - 1

Name - Samar Goyal
Roll no - 1024030992

Ans 2:

```
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 j = 0; j < n2; j++)
        R[j] = arr[mid + 1 + j];

    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) / 2;

        mergeSort(arr, left, mid);
        mergeSort(arr, mid + 1, right);

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

Output:

```
Sorted arr1: 5 6 7 11 12 13
Sorted arr2: 3 9 10 27 38 43 82
```

Assignment - 1

Name - Samar Goyal
Roll no - 1024030992

Ans 3:

```
int partition(int arr[], int low, int high) {
    int pivot = arr[high];   // pivot element
    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);
    }
}
```

Output:

```
Sorted array: 2 2 4 6 9 %
```

Assignment - 1

Name - Samar Goyal
Roll no - 1024030992

Ans 4:

```
int maxSubArraySum(int arr[], int n) {
    int maxSoFar = arr[0];
    int currentSum = arr[0];

    for (int i = 1; i < n; i++) {
        currentSum = max(arr[i], currentSum + arr[i]);
        maxSoFar = max(maxSoFar, currentSum);
    }

    return maxSoFar;
}
```

Output:

```
Maximum Subarray Sum = 7%
```