Quick Sort Pseudo Code –

Lowindex=0, highindex=arr.length-1

3 parameters passed to method int[] arr, int lowindex (first index) int highindex (last index)

3 variables declared int pivot = highindex, int left = lowindex, int right = highindex

Base Case

If(lowIndex>=highindex)

    return

While(left<right)

    While(arr[left]<=pivot and left<right)

        Left++

    While(arr[left]<=pivot and left<right)

        Right—

Swap(arr,left, right);

Helper Swap(int[] arr, int index1, index2)

Int temp = arr[index1]

Arr[index1]=arr[index2]

Arr[index]=temp;

Merge Sort Pseudo Code

Pass in the reference to the array.

Int length = arr.length

Int minIndex = length/2

Int[] leftHalf = new int[midindex]

Int rightHalf = new int[length-midindex]

For(i=0 I<midindex)

    Lefthalf[i]= arr[i]

For(i=midindex i<length)

    righthalf[i-index]= arr[i]

    mergeSort(leftHalf);

```
    mergeSort(rightHalf);
    merge(arr, leftHalf, rightHalf);
```

Helper Merge method

Pass in reference to original array left array and right array

Int leftsize and rightsize = the length of their array

Int i,j,k =0

```
While(i<leftSize && j<rightsize)
        if (leftHalf[i] <= rightHalf[j])
         arr[k] = leftHalf[i];
         i++
        else
         arr[k] = rightHalf[j];
         j++
while (i < leftSize)
      arr[k] = leftHalf[i];
      i++
      k++
    while (j < rightSize)
      arr[k] = rightHalf[j];
      j++;
      k++
```

Insertion Sort Pseudo Code

```
Pass in reference to array
Int curVal=0
For(i=1 i<arr.lenght )
curVal= arr[i]
int j=i-1
        while(j>= 0 && arr[j]>curVal)
                arr[j+1] = arr[j]
                j--
        arr[j+1]= curVal
```

Bubble Sort Pseudo Code

```
Pass in reference to array
Boolean isSwapped = true
While(isSwaped ==true)
        isSwapped = false
        for(i=0 i<arr.lenght-1)
                if(arr[i]>arr[i+1])
                        isSwapped = true
                        int temp = arr[i]
                        arr[i]=arr[i+1]
                        arr[i+1]=temp
```