Pseudo code

Code (recursive merge sort)
(algo4.c)

```
function merge(arr[], left, mid, right):
   n1 = mid - left + 1
   n2 = right - mid

   L[n1], R[n2]
     for i from 0 to n1-1:
      L[i] = arr[left + i]
   for j from 0 to n2-1:
      R[j] = arr[mid + 1 + j]

   i = 0
   j = 0
   k = left


   while i < n1 and j < n2:
      if L[i] <= R[j]:
         arr[k] = L[i]
         i++
      else:
         arr[k] = R[j]
         j++
      k++
   while i < n1:
      arr[k] = L[i]
      i++
      k++

   while j < n2:
      arr[k] = R[j]
      j++
      k++

function mergeSort(arr[], left, right):
   if left < right:
      mid = left + (right - left) / 2
```

```
        mergeSort(arr, left, mid)
        mergeSort(arr, mid + 1, right)
        merge(arr, left, mid, right)

function kthElement(a[], x, b[], y, k):
    n = x + y
    arr[n]

    for i from 0 to x-1:
        arr[i] = a[i]
    for i from 0 to y-1:
        arr[x + i] = b[i]
    mergeSort(arr, 0, n - 1)

    return arr[k - 1]

function main():
    arr1[] = {2, 5, 8, 1}
    arr2[] = {3, 6, 7, 0}
    x = kthElement(arr1, 4, arr2, 4, 6)
    print(x)

main()
```

Code (recursive insertion sort)
(algo.c)

```
function kthElement(a[], x, b[], y, k):
   n = x + y
   arr[n]     for i from 0 to n-1:
      if i >= x:
         arr[i] = b[i - x]
      else:
         arr[i] = a[i]
   for j from 0 to n-2:
      minIndex = j
      for i from j+1 to n-1:
         if arr[i] < arr[minIndex]:
            minIndex = i
      temp = arr[minIndex]
      arr[minIndex] = arr[j]
      arr[j] = temp

   return arr[k - 1]

function main():
   arr1[] = {2, 5, 8, 1}
   arr2[] = {3, 6, 7, 0}

   x = kthElement(arr1, 4, arr2, 4, 6)
   print(x)

main()
```

Code (non recursive master method)
(algo2.c)

```
Function funct(arr1,size1,arr2,size2,index){

       i=0,j=0,m=0,n=x+y

       Arr[n]

       while    m<n

                if    arr1[i]<arr2[i]

                         Arr[m]=arr1[i]

                         i++

               else

                         Arr[m]=arr2[j]

                          j++

              m++

      return  Arr[k-1]

}


main{

    arr1[],arr2[]

    numOfIndex=funct(arr1,size1,arr2,size2,index)

    print     (numOfIndex)
```

Code (recursive master method)
(algo3.c)

```
Function funt(arr1,size1,arr2,size2,index){

    if  size1=0

          return arr2[index-1]

    if   size2=0

          return arr1[index-1]

    if    arr1[0]<arr2[0]

           return  index==1 ? arr1[0] :  funct(arr1+1 ,size1-1,arr2,size2,index-1)

    else

           return  index==1 ? arr2[0] :  funct(arr1 ,size1,arr2+1,size2-1,index-1)

}

main{

   arr1[],arr2[]

  numOfIndex=funct(arr1,size1,arr2,size2,index)

  print   (numOfIndex)

}
```