**package** Question22;

**import** java.util.Arrays;

**import** java.util.Comparator;

**import** java.util.List;

**import** java.util.PriorityQueue;

// Data structure for Min Heap

**class** MinHeap

{

// Function to sort a K-Sorted Array

**public** **static** **void** sortKSortedArray(List<Integer> list, **int** k)

{

// create an empty min heap and insert first k+1 elements in the heap

PriorityQueue<Integer> pq = **new** PriorityQueue<>(list.subList(0, k+1));

**int** index = 0;

// do for remaining elements of the array

**for** (**int** i = k + 1; i < list.size(); i++)

{

// pop top element from min-heap and assign it to

// next available array index

list.set(index++, pq.poll());

// push next array element into min-heap

pq.add(list.get(i));

}

// pop all remaining elements from the min heap and assign it to

// next available array index

**while**(!pq.isEmpty())

{

list.set(index++, pq.poll());

}

}

**public** **static** **void** main(String[] args)

{

List<Integer> list = Arrays.*asList*(1, 4, 5, 2, 3, 7, 8, 6, 10, 9);

**int** k = 2;

*sortKSortedArray*(list, k);

System.***out***.println(list);

}

}