**import** java.util.Scanner;  
**public class** mayi{  
   
 **public static void** main(String[] args){  
 Scanner in = **new** Scanner(System.***in***);  
 **while**(in.hasNext()){  
 **int** n = in.nextInt();  
 **int** m = in.nextInt();  
 **int**[] num = **new int**[n];  
 **for**(**int** i = 0; i < n; i++)  
 num[i] = in.nextInt();  
 *quickSort*(num);  
 **for**(**int** i = 0;i < m - 1;i++){  
 System.***out***.print(num[i] + **","**);  
 }  
 System.***out***.println(num[m - 1]);  
 }  
 }  
 **public static void** quickSort(**int**[] Arrays) {  
  
 **if** (Arrays == **null** || Arrays.**length** < 2) {  
 **return**;  
 }  
 *quickSort*(Arrays, 0, Arrays.**length** - 1);  
 }  
 **public static void** quickSort(**int**[] Arrays, **int** l, **int** r) {  
 **if** (l < r) {  
 *swap*(Arrays, l + (**int**) (Math.*random*() \* (r - l + 1)), r);  
 **int**[] p = *partition*(Arrays, l, r);  
 *quickSort*(Arrays, l, p[0] - 1);  
 *quickSort*(Arrays, p[1] + 1, r);  
 }  
 }  
 **public static int**[] partition(**int**[] Arrays, **int** l, **int** r) {  
 **int** min = l - 1;  
 **int** max = r;  
 **while** (l < max) {  
 **if** (Arrays[l] < Arrays[r]) {  
 *swap*(Arrays, ++min, l++);  
 } **else if** (Arrays[l] > Arrays[r]) {  
 *swap*(Arrays, --max, l);  
 } **else** {  
 l++;  
 }  
 }  
 *swap*(Arrays, max, r);  
 **return new int**[] { min + 1, max };  
 }  
 **public static void** swap(**int**[] Arrays, **int** i, **int** j) {  
 **int** tmp = Arrays[i];  
 Arrays[i] = Arrays[j];  
 Arrays[j] = tmp;  
 }  
  
}

**基准三数取中切分**：

 package design;  
  
import java.util.ArrayList;  
import java.util.Scanner;  
  
/\*\*  
 \* 快排，3值取中  
 \*/  
public class kuaipai {  
        public static void main(String[] args) {  
            Scanner in = new Scanner(System.in);  
            while(in.hasNext()){  
  
                int n = in.nextInt();  
                int k = in.nextInt();  
                int[]input = new int[n];  
                for(int i = 0; i < n; i++){  
                    input [i] = in.nextInt();  
                }  
  
                ArrayList<Integer> list = new ArrayList<Integer>();  
                int L=input.length;  
                input = sort(input,0,L-1);  
                for(int i=0;i<L;i++){  
                    list.add(input[i]);  
                }  
                for(int i = 0;i<k;i++){  
                    System.out.println(list.get(i));  
                }  
            }  
  
        }  
  
        public static int [] sort(int [] input,int min,int max){  
            if(min>=max){  
                return input;  
            }  
            int index = quicksort(input,min,max);  
            sort(input,min,index-1);  
            sort(input,index+1,max);  
            return input;  
        }  
  
        public static int quicksort(int [] input, int min, int max){//每一次以基准去排  
            //int mid = dealPivot(input,min,max);  
  
            int Key =  dealPivot(input,min,max);  
            while(min<max){  
                while (Key<=input[max]&&min<max){//从后往前扫描  
                    max--;  
                }  
                input[min]=input[max];  
                input[max]=Key;  
                while(Key>=input[min]&&min<max){//从前往后扫描  
                    min++;  
                }  
                input[max]=input[min];  
                input[min]=Key;  
            }  
            input[max]=Key;  
            return max;  
        }  
  
        public static int dealPivot(int[] arr, int left, int right) {  
            int mid = (left + right) / 2;  
            if (arr[left] > arr[mid]) {  
                swap(arr, left, mid);  
            }  
            if (arr[left] > arr[right]) {  
                swap(arr, left, right);  
            }  
            if (arr[right] < arr[mid]) {  
                swap(arr, right, mid);  
            }  
            swap(arr, left, mid);  
            return arr[left];  
        }  
  
        private static void swap(int[] arr, int a, int b) {  
            int temp = arr[a];  
            arr[a] = arr[b];  
            arr[b] = temp;  
        }  
  
  
  
}