一、编程题

ACM：最大 N 个数与最小 N 个数的和

题目描述：给定一个数组，编写一个函数来计算它的最大N个数与最小N个数的和。你需要对数组进行去重。

说明：
*数组中数字范围[0, 1000]
*最大N个数与最小N个数不能有重叠，如有重叠，输入非法返回-1
*输入非法返回-1

输入描述：第一行输入M， M标识数组大小
第二行输入M个数，标识数组内容
第三行输入N，N表达需要计算的最大、最小N个数

输出描述：输出最大N个数与最小N个数的和。

补充说明：

示例1
输入：5
95 88 83 64 100
2
输出：342
说明：最大2个数[100,95],最小2个数[83,64], 输出为342

示例2
输入：5
3 2 3 4 2
2
输出：-1
说明：最大2个数[4,3],最小2个数[3,2], 有重叠输出为-1

代码：

```
import java.util.Scanner;
import java.util.Arrays;

// 注意类名必须为 Main, 不要有任何 package xxx 信息
public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);
        // 注意 hasNext 和 hasNextLine 的区别
        int m = in.nextInt();
        if (m == 0) {
            System.out.println(-1);
            return;
        }
        int[] nums = new int[m];
        for (int i = 0; i < m; i++) {
            nums[i] = in.nextInt();
        }
```

```
int n = in.nextInt();
if (in.hasNext()) {
        System.out.println(-1);
        return;
}
// while (in.hasNextInt()) { // 注意 while 处理多个 case
//         int a = in.nextInt();
//         int b = in.nextInt();
//         System.out.println(a + b);
// }

Arrays.sort(nums);
// for (int i = 0; i < m; i++) {
//         System.out.println(nums[i]);
// }

int nMin = 0;
int nMax = 0;
int cnt1 = 0;
int cnt2 = 0;
int pre = -1;
int ind1 = 0;
int ind2 = m - 1;
while (ind1 < m && cnt1 < n) {
        if (nums[ind1] != pre) {
                nMin += nums[ind1];
                //System.out.println(nums[ind1]);
                //pre = nums[ind1];
                cnt1++;
        }
        pre = nums[ind1];
        ind1++;
}
if (ind1 == m && cnt1 < n) {
        System.out.println(-1);
        return;
}

pre = -1;
while (ind2 >= 0 && cnt2 < n) {
        if (nums[ind2] != pre) {
                nMax += nums[ind2];
                //pre = nums[ind2];
                //System.out.println(nums[ind2])
```

```
                    cnt2++;
                }
                pre = nums[ind2];
                ind2--;
            }
            if (ind2 < ind1 - 1 || (ind2 < 0 && cnt2 < n)) {
                System.out.println(-1);
                return;
            }

            System.out.println(nMin + nMax);
            return;
        }
    }
```