一、编程题

ACM：内存冷热标记

题目描述：现代计算机系统中通常存在多级的存储设备，针对海量workload的优化的一种思路是将热点内存页优先放到快速存储层级，这就需要对内存页进行冷热标记。
一种典型的方案是基于内存页的访问频次进行标记，如果统计窗口内访问次数大于等于设定阈值，则认为是热内存页，否则是冷内存页。
对于统计窗口内跟踪到的访存序列和阈值，现在需要实现基于频次的冷热标记。内存页使用页框号作为标识。

输入描述：第一行为输入为N，表示访存序列的记录条数，$0 < N \leq 10000$。
第二行为访存序列，空格间隔的N个内存页框号，页框号范围0-65535，同一页框号可能重复出现，出现的次数即为对应页框号的频次。
第三行为热内存页的频次阈值T，正整数，范围$1 \leq T \leq 10000$。

输出描述：第一行输出标记为热内存的内存页个数，如果没有被标记为热内存的，则输出0。
如果第一行>0，则接下来按照访问频次降序输出内存页框号，一行一个，频次一样的页框号，页框号小的排前面。

补充说明：

示例1
输入：10
1 2 1 2 1 2 1 2 1 2
5
输出：2
1
2
说明：内存页1和内存页2均被访问了5次，达到了阈值5，因此热内存页有2个。内存页1和内存页2的访问频次相等，页框号小的排前面。

示例2
输入：5
1 2 3 4 5
3
输出：0
说明：访存跟踪里面访问频次没有超过3的，因此热内存页个数为0。

代码：

```
import java.util.*;
import java.util.stream.Collectors;

// 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 sc = new Scanner(System.in);
        int m = sc.nextInt();
        List<Integer> list = new ArrayList<>();
        for (int i = 0; i < m; i++) {
            list.add(sc.nextInt());
        }
        int sum = sc.nextInt();
        List<Page> pages = new ArrayList<>();
        for (Integer i : list) {
            List<Integer> pageList = pages.stream().map(Page::getPage).collect(Collectors.toList());
            Map<Integer, Page> pageMap = pages.stream().collect(Collectors.toMap(Page::getPage, item -> item));
            if (pageList.contains(i)) {
                Page page = pageMap.get(i);
```

```
                page.num = page.num + 1;
            } else {
                Page page = new Page(i);
                page.num = 1;
                pages.add(page);
            }
        }
        pages = pages.stream().filter(item -> item.num >= sum).collect(Collectors.toList());
        System.out.println(pages.size());
        if (pages.size() == 0) {
            return;
        }
        pages =
pages.stream().sorted(Comparator.comparing(Page::getPage)).sorted(Comparator.comparingInt(P
age::getNum).reversed()).collect(Collectors.toList());
        for (Page page : pages) {
            System.out.println(page.page);
        }
    }
}

class Page {
    int page;

    int num;

    public Page(int page) {
        this.page = page;
    }

    public int getPage() {
        return page;
    }

    public void setPage(int page) {
        this.page = page;
    }

    public int getNum() {
        return num;
    }

    public void setNum(int num) {
        this.num = num;
```

```
    }
}
```