

2022 CCF 非专业级软件能力认证模拟赛 - 入门组

2022/8/5

(请选手务必仔细阅读本页内容)

| | | | | |
|---------|-----------|------------|----------|-----------|
| 题目名称 | 试卷 | 魔法数字 | 排序 | 出栈 |
| 题目类型 | 传统型 | 传统型 | 传统型 | 传统型 |
| 英文题目名称 | paper | number | sort | stack |
| 输入文件名 | paper.in | number.in | sort.in | stack.in |
| 输出文件名 | paper.out | number.out | sort.out | stack.out |
| 输出文件名 | 1s | 2s | 1s | 1s |
| 内存上限 | 512M | 512M | 512M | 512M |
| 测试点数目 | 10 | 10 | 10 | 10 |
| 每个测试点分值 | 10 | 10 | 10 | 10 |
| 附加样例文件 | 有 | 有 | 有 | 有 |
| 结果比较方式 | 全文比较 | 全文比较 | 全文比较 | 全文比较 |

注意事项:

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回类型必须是 `int`，程序正常结束时的返回值必须是 `0`。

1 试卷

(paper.cpp/c)

【题目描述】

藤藤正在准备计算机专业能力考试试题，认真负责的他精心挑选了 k 道试题（编号 $1 \sim k$ ）组成了题库，并由系统随机抽选其中 n 题作为考试试卷。

系统在组卷时自动组成了 A 卷和 B 卷，其中 A 卷 n 道题的编号为 a_1, a_2, \dots, a_n ，其中 B 卷 n 道题的编号为 b_1, b_2, \dots, b_n 。

请你帮藤藤计算一下，有多少道试题同时出现在 A, B 卷中，并按从小到大的顺序给出重复试题的编号。

【输入输出格式】

【输入格式】

从文件 paper.in 中读入数据。

输入共三行：

第一行，两个正整数表示 n, k

第二行， n 个正整数表示 a_1, a_2, \dots, a_n

第三行， n 个正整数表示 b_1, b_2, \dots, b_n

【输出格式】

输出到文件 paper.out 中。

输出第一行：重复试题的数量

输出第二行：按从小到大顺序输出重复试题的编号

（若第一行重复试题的数量为 0，则不用输出第二行的信息）

【样例】

【样例 1 输入】

```
5 10
1 2 3 4 5
10 9 8 7 6
```

【样例 1 输出】

```
0
```

【样例 2 输入】

5 10
6 4 1 8 10
5 1 7 4 6

【样例 2 输出】

3
1 4 6

【数据范围与提示】

对于 30% 的数据, $1 \leq k \leq 10$

对于 70% 的数据, $1 \leq k \leq 10^4$

对于 100% 的数据, $1 \leq k \leq 10^5$

$1 \leq n \leq k, 1 \leq a_1, a_2, \dots, a_n \leq k, 1 \leq b_1, b_2, \dots, b_n \leq k$

2 魔法数字

(number.cpp/c/pas)

【题目描述】

自然数在 10 进制下，其 **正整数** 次幂的尾数等于它自己的数即为魔法数字。

比如 5 就是魔法数字，因为 $5^1 = 5$, $5^2 = 25$, $5^3 = 125 \dots$

藤藤对魔法数字很感兴趣，便找来了你，请你帮他找出所有 K 以内 (含 K) 的魔法数字

【输入输出格式】

【输入格式】

输入文件 number.in

输入仅一行，为题目中的 K 。

【输出格式】

输出文件 number.out

输出仅一行，为 **所有** K 以内 (含 K) 的魔法数字，从小到大排序，数与数之间用空格隔开。

【样例】

【样例 1 输入】

20

【样例 1 输出】

0 1 5 6

【数据范围与提示】

对于 30% 的数据， $0 \leq K \leq 4 \times 10^4$;

对于 50% 的数据， $0 \leq K \leq 5 \times 10^7$;

对于 70% 的数据， $0 \leq K \leq 3 \times 10^9$;

对于 100% 的数据， $0 \leq K \leq 10^{19}$ 。

3 排序

(sort.cpp/c/pas)

【题目描述】

给定一个长度为 n 的排列，你可以：

- 选择一个数字，花费 L 点体力，将其向左移动任意多个位置
 - 选择一个数字，花费 R 点体力，将其向右移动任意多个位置
- 计算一下，为使整个排列升序，最小需要花费多少点体力？

【输入输出格式】

【输入格式】

从文件 sort.in 中读入数据

输入第一行，一个正整数 n

输入第二行， n 个正整数 p_1, p_2, \dots, p_n ，表示给定排列

输入第三行，两个正整数 L, R ，分别表示向左向右移动数字所花费的体力值。

【输出格式】

输出到文件 sort.out

输出共一行，表示为使整个排列升序，最小需要花费的体力值

【样例】

【样例 1 输入】

```
4
1 4 2 3
2 3
```

【样例 1 输出】

```
3
```

若选择数字 2 向左移动至数字 4 前，代价为 2，再选择数字 3 向左移动至数字 4 前，代价为 2，总代价为 4

若选择数字 4，向右移动至末尾，代价为 3

【样例 2 输入】

```
3
1 2 3
3 5
```

【样例 2 输出】

```
0
```

【数据范围与提示】

对于 30% 的数据, $1 \leq n \leq 10$

对于 60% 的数据, $1 \leq n \leq 10^3$

对于 100% 的数据, $1 \leq n \leq 10^4, 1 \leq L, R \leq 10^9$

4 出栈

(stack.cpp/c/pas)

【题目描述】

现在有一个栈。

给出一个正整数 n ，以及一个长度为 n 的进栈序列。序列保证是一个 $1 - n$ 的排列。

每次都可以选择进行以下两种操作之一，直至两种操作都不能进行：

1. 将当前进栈序列最前面的数入栈。
2. 将当前栈顶的数出栈。

合法的操作定会产生出一组出栈序列，请你求出字典序最大的出栈序列。

【输入输出格式】

【输入格式】

从文件 stack.in 中读入数据

第一行一个正整数 n 。

第二行 n 个整数，表示入栈序列。

【输出格式】

输出到文件 stack.out 中

输出 n 个整数，表示字典序最大的出栈序列。

注意：滑滑梯总部不算占领的星球

【样例】

【样例 1 输入】

```
5
2 1 5 3 4
```

【样例 1 输出】

```
5 4 3 1 2
```

样例解释：2 入栈；1 入栈；5 入栈；5 出栈；3 入栈；4 入栈；4 出栈；3 出栈；1 出栈；2 出栈。

【样例 2 输入】

```
3
3 2 1
```

【样例 2 输出】

3 2 1

【数据范围与提示】

对于 50% 的数据， $n \leq 50$ 。

对于 70% 的数据， $n \leq 10^3$ 。

对于 100% 的数据， $1 \leq n \leq 10^5$ 。