

第 24 届全国青少年信息学奥林匹克联赛

CCF-NOIP-2018

附中校园桂花飘香@真模拟赛

题目名称	击鼓传花	排序	火车车厢	机器翻译
题目类型	传统型	传统型	传统型	传统型
目录	flower	bigsort	train	translate
可执行文件名	flower	bigsort	train	translate
输入文件名	flower.in	bigsort.in	train.in	translate.in
输出文件名	flower.out	bigsort.out	train.out	translate.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒	1.0 秒
内存限制	256MB	256MB	256MB	256MB
测试点数目	20	10	20	10
每个测试点分值	5	10	5	10

注意事项：

- 1、文件名（程序名及输入输出文件名）必须使用英文小写
- 2、以“学校+姓名”建立文件夹，把 cpp 文件放入其中
- 3、整体提交该文件夹，无需再建子目录
- 4、每个程序都需要使用文件输入输出
- 5、除非特别说明，结果比较方式均为忽略行末空格及文末回车的全文比较
- 6、编译时不打开任何优化选项

1. 击鼓传花 (flower.cpp)

【问题描述】

有 n 个人围成一个圈，编号按照顺时针分别是 $1, 2, \dots, n$ ，然后 1 在 n 的顺时针方向。一开始编号为 x 的人手上有一朵花，手上有花的人每次都会从他本人开始，往他的顺时针方向数 k 个人，然后把花给到这个人手上，被给到花的这个人也会进行同样的操作，之后依次类推。

问编号为 t 的人是否可能拿到花，如有，输出 “Yes”，否则输出 “No”。

【输入格式】

从文件 `flower.in` 中读入数据。

输入仅一行，四个正整数 n, k, x, t ，意义如题所述。

【输出格式】

输出到文件 `flower.out` 中。

输出仅一行一个串，“Yes” 表示有机会，“No” 表示没有机会（不含引号）

【样例输入】

9 6 1 4

【样例输出】

Yes

【样例输入 2】

10 6 1 4

【样例输出 2】

No

【样例解释】

样例 1: 1 -> 7 -> 4, 4 号位可以拿到花

样例 2: 1 -> 7 -> 3 -> 9 -> 5 -> 1 -> 7 -> ... , 4 号位永远也拿不到花

【数据范围】

对于 100% 的数据, $n, k \leq 10^9$

2. 排序 (bigsort.cpp)

【问题描述】

给定 n 个长为 20 的数列, 对这些数列从小到大进行排序。

定义数列大小比较方式为: 先比较首项, 首项小的数列小; 首项一样则比较第二项, 以此类推。

【输入格式】

从文件 bigsort.in 中读入数据。

输入文件共有 $n+1$ 行, 第一行一个整数 n 。

接下来 n 行, 每行 20 个用空格隔开的 0~9 的正整数, 一个数列。

【输出格式】

输出到文件 bigsort.out 中。

输出文件共有 n 行, 每一行是 20 个用空格隔开的 0~9 的正整数, 表示一个数列。

【样例输入】

5

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 4
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 5
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 8
```

【样例输出】

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 5
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 4
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 8 2
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 9 8
```

【数据范围】

对于 100%的数据， $n \leq 10^5$

3. 火车车厢 (train.cpp)

【问题描述】

在一个火车站旁边有一座桥，其桥面可以绕河中心的桥墩水平旋转。一个车站的职工发现桥的长度最多能容纳两节车厢，如果将桥旋转 180 度，则可以把相邻两节车厢的位置交换，用这种方法可以重新排列车厢的顺序。

于是他就负责用这座桥将进站的车厢按车厢号从小到大排列。他退休后，火车站决定将这一工作自动化，其中一项重要的工作是编一个程序，输入初始的车厢顺序，计算最少用多少步就能将车厢排序

【输入格式】

从文件 train.in 中读入数据。

输入数据有两行，第一行一个整数 n 。

第二行， n 个用空格隔开的正整数 a_i 。

【输出格式】

输出到文件 train.out 中。

输出仅一行，一个正整数，即答案。

【样例输入】

```
4
3 2 3 2
```

【样例输出】

```
3
```

【数据范围】

对于 100% 的数据， $a_i \leq 10^5$

4. 机器翻译

(`translate.cpp`)

【问题描述】

某翻译软件原理很简单，它从头到尾，依次将每个英文单词用对应的中文含义来替换。对于每个英文单词，软件会先在内存中查找这个单词的中文含义，如果内存中有，软件就会用它进行翻译；如果内存中没有，软件就会在外存中的词典内查找，查出单词的中文含义然后翻译，并将这个单词和译义放入内存，以备后续的查找和翻译。

假设内存中有 M 个单元，每单元能存放一个单词和译义。每当软件将一个新单词存入内存前，如果当前内存中已存入的单词数不超过 $M-1$ ，软件会将新单词存入一个未使用的内存单元；若内存中已存入 M 个单词，软件会清空最早进入内存的那个单词，腾出单元来，存放新单词。

假设一篇英语文章的长度为 N 个单词。给定这篇待译文章，翻译软件需要去外存查找多少次词典？假设在翻译开始前，内存中没有任何单词。

【输入格式】

从文件 `translate.in` 中读入数据。

第一行两个整数 m 、 n 。分别表示内存容量和文章长度。

第二行 n 个整数 w_i ，按照文章顺序，每个 w_i 代表一个英文单词。文章中两个单词是同一个，当且仅当它们对应的整数相同。

【输出格式】

输出到文件 `translate.out` 中。

一个整数，为该翻译软件需要查词典的次数。

【样例输入】

```
3 7
1 2 1 5 4 4 1
```

【样例输出】

5

【样例输入 2】

2 10

8 824 11 78 11 78 11 78 8 264

【样例输出 2】

6

【数据范围】

对于 100% 的数据， $m \leq 100$ ， $n \leq 1,000$ ， $w_i \leq 1,000$