

南京市小学生信息学竞赛初赛编程模拟卷（一）

姓名：

成绩：

1. 明明的随机数 (random.cpp)

【问题描述】

明明想在学校中请一些同学一起做一项问卷调查，为了实验的客观性，他先用计算机生成了 N 个 1 到 1000 之间的随机整数 ($N \leq 100$)，对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助明明完成“去重”与“排序”的工作。

【输入文件】

输入文件 random.in 有 2 行，第 1 行为 1 个正整数，表示所生成的随机数的个数：

N

第 2 行有 N 个用空格隔开的正整数，为所产生的随机数。

【输出文件】

输出文件 random.out 也是 2 行，第 1 行为 1 个正整数 M ，表示不相同的随机数的个数。第 2 行为 M 个用空格隔开的正整数，为从小到大排好序的不相同的随机数。

【输入样例】

```
10
20 40 32 67 40 20 89 300 400 15
```

【输出样例】

```
8
15 20 32 40 67 89 300 400
```

2. 机器翻译 (translate.cpp)

【问题描述】

小晨的电脑上安装了一个机器翻译软件，他经常用这个软件来翻译英语文章。

这个翻译软件的原理很简单，它只是从头到尾，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】

输入文件名为 `translate.in`，输入文件共 2 行。每行中两个数之间用一个空格隔开。

第一行为两个正整数 M 和 N ，代表内存容量和文章的长度。

第二行为 N 个非负整数，按照文章的顺序，每个数（大小不超过 1000）代表一个英文单词。文章中两个单词是同一个单词，当且仅当它们对应的非负整数相同。

【输出】

输出文件 `translate.out` 共 1 行，包含一个整数，为软件需要查词典的次数。

【输入输出样例 1】

<code>translate.in</code>	<code>translate.out</code>
3 7	5
1 2 1 5 4 4 1	

【输入输出样例 1 说明】

整个查字典过程如下：每行表示一个单词的翻译，冒号前为本次翻译后的内存状况：

空：内存初始状态为空。

1. 1: 查找单词 1 并调入内存。
2. 1 2: 查找单词 2 并调入内存。
3. 1 2: 在内存中找到单词 1。
4. 1 2 5: 查找单词 5 并调入内存。
5. 2 5 4: 查找单词 4 并调入内存替代单词 1。
6. 2 5 4: 在内存中找到单词 4。
7. 5 4 1: 查找单词 1 并调入内存替代单词 2。

共计查了 5 次词典。

【输入输出样例 2】

<code>translate.in</code>	<code>translate.out</code>
2 10	6
8 824 11 78 11 78 11 78 8 264	

【数据范围】

10% 的数据： $M=1$ ， $N \leq 5$ 。

100% 的数据： $0 < M \leq 100$ ， $0 < N \leq 1000$ 。

3. 单词匹配 (`match.cpp`)

【问题描述】

为了更好地记住英语单词，我们经常需要比较两个单词的相似程度。如何准确地衡量两个单词的相似程度呢？

我们想了个办法，举例来说：对于单词 `CAPILLARY` 和 `MARSUPIAL`，我们可以这样来比较它们：

`CAPILLARY`

`MARSUPIAL`

这样只有一个字母 (A) 一样，更好的方法是这样对齐：

`CAPILLARY`

MARSUPIAL

则它们就有两个字母一样了（A和R），但最好的方法是这样：

CAPILLARY

MARSUPIAL

因为这样，它们就有3个字母一样（P，I和L）。这也是这两个单词的最大可能的相似长度了。

根据以上方法，我们可以定义以下一个公式来准确地衡量两个单词的相似程度，假设单词为word1和word2，它们的长度分别为length1和length2，它们的最大相似长度为length，则我们用以下分数来表示它们的相似程度：

$$\frac{\text{length} * 2}{\text{length1} + \text{length2}}$$

于是，CAPILLARY 和 MARSUPIAL 这两个单词的相似程度为 $3*2/(9+9) = 1/3$ 。显然两个单词最大相似程度为 1（即两个单词是一样的），最小相似程度为 0。

【输入】(match.in)

若干行(最多 1000 行)，每行为一组数据，每组数据是用一个空格隔开的两个单词，每个单词最长 100 个字符，且都是大写字母。

【输出】(match.out)

对于每一行输入，输出一个最简分数，如果是 1/1 就输出 1，如果是 0/1 就输出 0。

【输入样例】

CAR CART

TURKEY CHICKEN

MONEY POVERTY

【输出样例】

6/7

4/13

1/3

4. 车厢重组 (carry.cpp)

【问题描述】

在一个旧式的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】(carry.in)

输入文件有两行数据，第一行是车厢总数 N（不大于 10000），第二行是 N 个不同的数表示初始的车厢顺序。

【输出】(carry.out)

一个数据，是最少的旋转次数。

【输入样例】

4

4 3 2 1

【输出样例】

6

5. 取数游戏(choice.cpp)

【问题描述】

我们来玩一个游戏：将自然数 1 到 N ，按顺序列成一排，你可以从中取走任意个数，但是相邻的两个不可以同时被取走。请你算出一共有多少种不同的取法。

【输入】(choice.in)

一个数 n 。

【输出】(choice.out)

一个数，表示不同取法的种数。

【输入样例】

2

【输出样例】

3

【样例说明】

3 种取法分别是：取 1，取 2，或一个也不取。

【数据范围】

60%的数据满足： $1 \leq n \leq 40$ ；

100%的数据满足： $1 \leq n \leq 1000$ 。