

# 测试

高天宇

编译命令	g++ -o * *.cpp gcc -o * *.c fpc *.pas
评测软件	Lemon
评测方式	忽略行末空格和回车

## A 消失的数字

文件名	输入文件	输出文件	时间限制	空间限制
del.cpp/c/pas	del.in	del.out	1s	512MB

### 题目描述

现在，我的手上有  $n$  个数字，分别是  $a_1, a_2, a_3, \dots, a_n$ 。

我现在需要删除其中的  $k$  个数字。当然我不希望随随便便删除，我希望删除  $k$  个数字之后，剩下的  $n - k$  个数中有最多的不同的数。

### 输入格式

第一行两个正整数  $n$  和  $k$ ，含义如题目描述。

接下来一行，有  $n$  个非负整数，分别是  $a_1$  到  $a_n$ 。

### 输出格式

一共一行，一个整数  $ans$ ，表示删除了  $k$  个数字后最多的不同的数的个数。

### 样例输入

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

### 样例输出

```
3
```

### 样例解释

如果删去第一个 1:

在  $[3, 1, 2]$  中有 3 个不同的数

如果删去 3:

在  $[1, 1, 2]$  中有 2 个不同的数

如果删去第二个 1:

在  $[1, 3, 2]$  中有 3 个不同的数

如果删去 2:

在  $[1, 3, 1]$  中有 1 个不同的数

## 数据范围

对于 30% 的数据,  $n \leq 10$ ,  $a_i \leq 10$ 。

对于 60% 的数据,  $n \leq 100$ ,  $a_i \leq 100$ 。

对于 80% 的数据,  $n \leq 10^5$ ,  $a_i \leq 10^5$ 。

对于 100% 的数据,  $n \leq 10^5$ ,  $a_i \leq 10^9$ 。

## B 国际跳棋

文件名	输入文件	输出文件	时间限制	空间限制
chess.cpp/c/pas	chess.in	chess.out	1s	512MB

### 题目描述

国际跳棋是一种古老的棋类游戏，远在古埃及法老时期就已存在，现代国际跳棋是在 12 世纪定型的。国际跳棋是由各国的民族跳棋演变而来，其历史源远流长。简化版（与标准国际跳棋略有差别）国际跳棋规则：

- 国际跳棋的棋盘由  $10 \times 10$  共 100 个格子组成
- 初始的时候，黑白双方各有 20 个棋子
- 移动：可以将我方任意棋子向左前方或右前方移动 1 步
- 跳吃：只要左前方、右前方、左后方、右后方相邻格子有对方棋子，且跳过这枚对方棋子后有空位，则可以跳过对方棋子并将对方棋子吃掉。如你的棋子在  $(x, y)$ ，对方棋子在  $(x+1, y+1)$ ， $(x+2, y+2)$  为空，则你可以跳到  $(x+2, y+2)$  并吃掉对方的棋子
- 加冕：任何一个棋子在行动过程停止的时候停到了对方底线（最靠近对方的一行）就可以加冕，从此成为“王”。注意，连续跳吃的时候只有最后一步停在对方底线才可以加冕
- 连跳：跳吃可以由多次跳吃组成。
- 王的特权：王在移动的时候可以无视方向（左前、右前、左后、右后都可以），无视距离（走几步都行，直到遇到别的棋子），无视跳吃距离（比如说  $(x, y)$  跳过  $(x+3, y+3)$  落到  $(x+7, y+7)$  是可以的，但是这中间除了有被吃掉的对方棋子，不能有其他棋子
- 在跳吃结束的时候才将被吃掉的棋子拿出棋盘，在这之前作为“屏障”，即这些棋子不能再次被跳吃，也不能落子
- 按照以上规则，给定一个棋局，合法的操作方案有很多。然而，每次必须选择吃子最多的操作方案。比如，在某种棋局下，有 A、B、C、D 四种方案，A、B 吃子 3 枚，C 吃 1 枚，D 吃 0 枚，则真正合法的操作总数为 2

作为一个国际跳棋迷，陶陶想要编写一个网络对战跳棋软件。然而他现在不会判断怎样的操作是合法的。对于给定的局面，你能给出所有合法的操作吗？

## 输入格式

输入数据是两个十行十列的矩阵，第一个矩阵中的每个点可能是以下三种：

- 0 空位置
- 1 我方棋子
- 2 对方棋子

第二个矩阵描述的是国王的情况。若为 1，表示是国王；为 0 表示不是国王。

## 输出格式

输出第一行为一个数字，表示合法操作的个数  $ans$ 。

下面一共  $ans$  行，每行表示一种合法操作中被操作的棋子。格式为  $(x, y)$  表示该棋子在第  $x$  行、第  $y$  列（注意，逗号后面没有空格）。如果某一个棋子有多种合法操作，则输出多遍。输出的顺序按从上到下、从左到右。

如果没有任何合法操作，只输出一个 0 即可

## 样例输入 1

```
0000000000
0000100000
0000000200
0000100000
0000000200
0000001000
0000000200
2000000000
0101000200
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
```

0000000000

0000000000

0000000000

0000000000

### 样例输出 1

2

(6,7)

(6,7)

## C 天上掉馅饼

文件名	输入文件	输出文件	时间限制	空间限制
bonus.pas/c/cpp	bonus.in	bonus.out	1s	128MB

### 题目描述

小 G 进入了一个神奇的世界，在这个世界，天上会掉下一些馅饼。今天，天上会随机掉下  $k$  个馅饼。

每次天上掉下馅饼，小 G 可以选择吃或者不吃（必须在下一个馅饼掉下来之前作出选择，并且现在决定不吃的话以后也不能吃）。

馅饼有  $n$  种不同的馅，根据物理定律，天上掉下这  $n$  种馅饼的概率相同且相互独立。然而，每一种馅饼  $i$  都有一个前提馅饼集合  $S_i$ 。只有当  $S_i$  中的馅饼都吃过之后，才能吃第  $i$  种馅饼。比如说，韭菜馅馅饼的  $S$  中有白菜猪肉馅饼和鲜虾馅饼，那么小 G 只有在吃过白菜猪肉馅饼和鲜虾馅饼之后，才能吃韭菜馅的馅饼。

同时，每个馅饼还有一个美味值  $P_i$ 。今天一天小 G 的幸福度，等于小 G 吃到的所有馅饼的美味值之和。注意， $P_i$  可能是负数。

现在考虑，在采用最优策略的前提下，小 G 这一天期望的幸福度是多少？

### 输入格式

第一行两个正整数  $k$  和  $n$ ，表示馅饼的数量和种类。

以下  $n$  行，每行若干个数，描述一种馅饼。其中第一个数代表美味值，随后的整数表示该馅饼的前提馅饼，以 0 结尾。

### 输出格式

输出一个实数，保留 6 位小数，即在最优策略下期望的幸福度。

### 样例输入 1

```
1 2
1 0
2 0
```

### 样例输出 1

```
1.500000
```

## 数据范围

对于 20% 的数据，所有的馅饼都没有“前提馅饼”

对于 50% 的数据， $1 \leq k \leq 10$ ， $1 \leq n \leq 10$

对于 100% 的数据， $1 \leq k \leq 100$ ， $1 \leq n \leq 15$ ，美味度为属于  $[-10^6, 10^6]$  的整数