

## 目录

问题 A 咱俩下象棋去.....	2
问题 B 抽积木游戏.....	5
问题 C 梅贾的窃魂卷.....	7
问题 D 强行数学题.....	8
问题 E 羊村保卫站.....	9
问题 F 《特殊三角形》 .....	10
问题 G 套马的汉子.....	12
问题 H 繁忙的路口.....	14
问题 I roll up.....	16
问题 J 羊村快递站.....	17
问题 K 推箱子.....	19
问题 L Alice and Bob.....	21
问题 M 简单字符串匹配.....	22

# 问题 A 咱俩下象棋去

## 描述：

ysj 和 lzl 正在下象棋，这时，ysj 说了一句将军，但是懂得象棋规则的 wtj 觉得 ysj 并没有将军，并且大肆嘲笑了 ysj。

然后就被 lzl 和 ysj 一起喷了，原来他们下的不是象棋，而是他们自创的一种棋。用的是象棋的规则和棋子。但是**这种棋是可以选择一个棋子，连续走两步。**

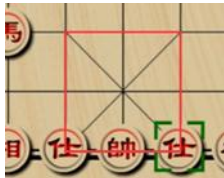
现在给你这个棋盘的情况，请你告诉 wtj，ysj 是否将军了。（不需要考虑当前棋盘的情况是否合理，只需要判断 ysj 是否将军了，ysj 使用的是**红棋**）。

如果 ysj 将军了，输出“yes”，否则输出“no”。（不含引号）

规则如下：（规则同普通象棋规则，只是每次轮到某一方下的时候，他可以选择**同一个棋子**连续下两步，如果你已经知道象棋规则可以跳过下面的规则）

H：表示没有棋子

K：表示将军和帅。S：表示士。这两者都只能在下图框柱的范围内移动，不同的是士只能走斜线，帅只能走直线。



B：表示兵，每步只能走一个格子，并且在没有过**楚河汉界**的时候只能往前走，过了之后能左右走，也可以往前走。

C：表示车，走直线，格数不限制，即可以走到四个方向的任意位置，如果有棋子挡住了则最多行走到挡住的棋子的位置。

P：表示炮，炮走直线，格数不限。如果炮想吃对方的棋子，则需要中间有一个棋子（不论颜色），才能跳到**对方棋子**位置，并且吃掉对方棋子。只需要中间有一个棋子就可以跳过该棋子，且距离不限。

### 中间有且仅有一个棋子

如图红色椭圆是跑能走到的位置，炮之所以能走到黑棋的卒上，是因为中间有一个兵，且只有这一个兵。

不能跳到黑棋炮的位置，是因为这样两个炮之间有两个棋子，不符合规则。如果卒这个位置没有棋子，则可以跳到黑棋炮的位置。

因为这样两个炮之间只有一个棋子，虽然红棋炮到兵的距离为 2，黑棋炮到兵的距离为 3，但是炮的距离是无限的，唯一要求就是中间有一个棋子。



X: 表示象，只能走田字格，**不能过楚河汉界**。田字格即是走 2\*2 的范围。

但是如果要走到的位置和原来位置对角线上有棋子的话，就不能走过去。即图中黑色椭圆圈起来的位置，就是因为车挡住了，所以不能走到，而红色椭圆圈起来的位置能走到。



M: 表示马，只能走日字格，如图



我们可以把马的走动理解为，先向要走 2 个格子距离的方向走，再走一个格子的距离。所以如果要走 2 个格子距离的方向上有棋子挡住了，就不能行走。

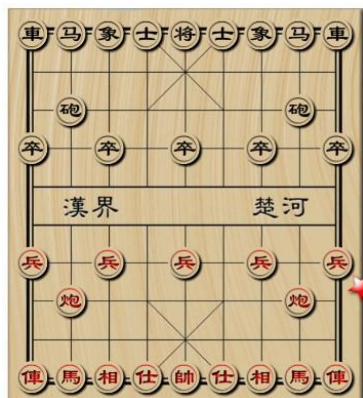
黑色椭圆圈起来的位置就是本来能走，但是被其他棋子挡住了，所以不能走到的位置。

红色椭圆圈起来的位置是能走到的位置。

黄色椭圆圈起来的位置是本来能走，但是该位置上是友方棋子所以不能走到的位置。

**棋子想要走到的地方，如果有己方棋子，也不能走到该位置，如果有敌方棋子，则吃掉敌方棋子。**

所以下列棋盘的情况是这样的，小写字母为红棋，大写字母为黑棋。



```
C M X S K S X M C
H H H H H H H H H
H P H H H H H P H
B H B H B H B H B
H H H H H H H H H
H H H H H H H H H
b H b H b H b H b
H p H H H H H p H
H H H H H H H H H
c m x s k s x m c
```

### 输入：

输入 10 行 每行 9 个字符表示当前位置棋子情况，每个字符用空格隔开。没有行末空格。

### 输出：

输出 ysj 是否将军（如果 ysj 再走一步棋或者两步棋就能吃掉 lzl 的帅，那么我们认为 ysj 将军了，不考虑帅和将见面的情况）

### 样例输入：

```
C M X S K S X M C
H H H H H H H H H
H P H H H H H P H
B H B H B H B H B
H H H H H H H H H
H H H H H H H H H
b H b H b H b H b
H p H H H H H p H
H H H H H H H H H
c m x s k s x m c
```

### 样例输出：

no

## 问题 B 抽积木游戏

### 描述：

你正在玩一个抽积木游戏。

刚开始所有积木按顺序从下到上堆积，每个积木有个大小。

这个抽积木游戏有个特殊的规律，你发现如果积木的长度能够被（此积木所在的高度+1）整除，则能被安全抽出，否则不能。

高度从 1 到  $n$  排列。当你抽走一个积木的时候，这个积木上面所有积木都按顺序往下掉 1，即高度减 1。

按下到上的顺序给你  $n$  个积木的大小，问你能不能存在某种顺序安全抽完所有的积木。

**注意：**只剩一个积木也需要考虑能否安全抽出，即这个积木的大小能不能被 2 整除。最上面的积木也得考虑安全问题。

例如：积木摆放

{2}可以被安全抽完

{4,6}可以被安全抽完 先抽 4，再抽 6

{6,9}可以被安全抽完 先抽 9，再抽 6

{3,2}不可以被抽完

{3}不可以被抽完

### 输入：

第一行一个数  $n(1 \leq n \leq 2e6)$  代表积木的数量

第二行  $n$  个数代表积木的摆放，从低到高的顺序给出，积木的大小是一个正整数  $a$  ( $1 \leq a \leq 2e6$ )

### 输出：

YES 代表可以抽完

NO 代表不能抽完

### 样例输入：

5

16 6 12 9 36

### 样例输出：

YES

## 提示

16 6 12 9 36

先抽走 36  $36/(5+1)=6$

16 6 12 9

再抽走 6  $6/(2+1)=2$

16 12 9

再抽走 12  $12/(2+1)=4$

16 9

再抽走 9  $9/(2+1)=3$

16

最后抽走 16  $16/(1+1)=8$

所以可以抽完，输出 YES

## 问题 C 梅贾的窃魂卷

### 描述：

Scout 在开局 7 分钟就买了一本梅贾的窃魂卷，俗称“杀人书”，杀人书（一共 25 层被动）的功效是击杀一名敌方英雄就增加 5 层被动，获得一次助攻会增加 2 层被动，而被击杀一次会掉 10 层被动，掉到 0 层不会再掉，scout 杀人书初始为 0 层。现在已知在游戏结束时 scout 的杀人书为 25 层，给出 scout 的击杀数  $k$ ，死亡数  $d$ ，以及助攻数  $a$ ，请问这三种字母共有多少种出现顺序能让 scout 最后的杀人书为 25 层，超过 25 层也为 25 层。

### 输入：

输入三个整数  $k(0 < k \leq 10)$ ,  $d(0 \leq d \leq 5)$ ,  $a(0 < a \leq 5)$

### 输出：

输出一个整数，符合题意的排列数

### 样例输入：

3 1 5

### 样例输出：

56

### 提示

样例中 dkkkaaaaa 和 dkakakaaa 为不同的两种

## 问题 D 强行数学题

**描述：**

议员发现两个有趣的式子

$$\frac{1}{1} + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \cdots = \frac{\pi \cdot \pi}{6}$$

$$\frac{1}{1} + \frac{1}{9} + \frac{1}{25} + \frac{1}{49} + \frac{1}{81} + \cdots = \frac{\pi \cdot \pi}{8}$$

但是议员现在满脑子都是黑妹，因此他算不出下面这个式子

$$\frac{1}{1} - \frac{1}{4} + \frac{1}{9} - \frac{1}{16} + \frac{1}{25} - \cdots = ?$$

于是把这个问题交给了你们来计算，结果仅保留 3 位小数

**输入：**

无

**输出：**

无,结果仅保留 3 位小数

**样例输入：**

无

**样例输出：**

无



## 问题 E 羊村保卫站

### 描述：

美丽的青青草原上生长着肥美的绿草，养育着羊村一村羊，但是由于温室效应加剧，整个青青草原缩小到只剩一块正方形区域以  $(x_0, y_0)$  为左上角以  $d$  为边长，羊村一村羊也被迫生活在其中，这时灰太狼觉得机会来了，于是想出了一条妙计：以狼堡为圆心用绳子牵一头牛，把青青草原的草都给干完，羊们无处可去便会落入狼口。但是绳子就快不够了，灰太狼也由于太久没吃羊精神恍惚，不知道至少该用多长的绳子才能把青青草原的草干完，你作为灰太狼亲爱的侄子，快帮你叔叔算算吧？

狼堡和牛都可以看作质点。

### 输入：

第一行输入三个整数， $x_0$ ， $y_0$ ， $d$ ，代表青青草原的左上角和边长。

第二行输入两个整数， $x_1$ ， $y_1$ ，代表狼堡的位置。

### 输出：

输出一个数代表所需的最短的绳子长度（保留三位小数）。

### 样例输入：

0 0 2

2 2

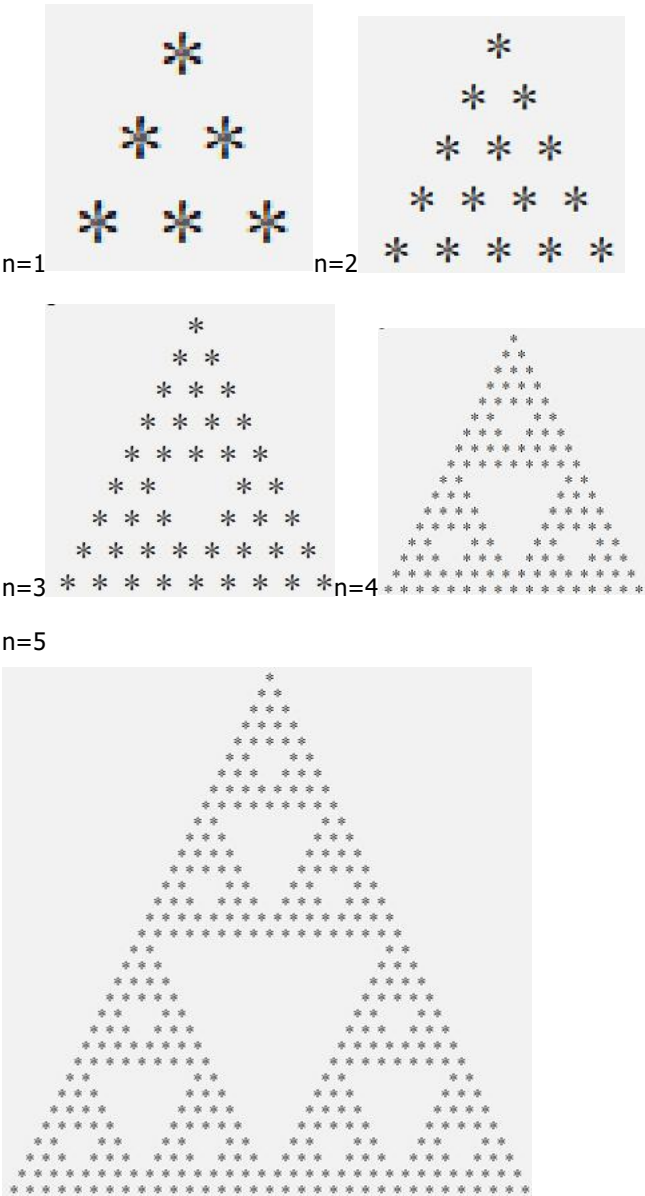
### 样例输出：

2.828

# 问题 F 《特殊三角形》

描述：

根据以下规律打印图形



输入：

输入一个正整数 n (1<=n<=8)

输出：

打印 n 对应的图形(注意，输出答案图形前空一行，**第一列全是空格，也就是最左边一列全是空格，行尾无多余空格**)

**样例输入：**

(见描述)

**样例输出：**

(见描述)

# 问题 G 套马的汉子

## 描述：

一天 **ysj** 写出来了一个题目，在实验室里欢呼，**wtj** 看到他这么高兴，就过去看了看他写了个什么题目。题目如下

如果矩阵 **A** 中存在这样的元素 **A[i,j]** 满足条件: **A[i,j]** 是第 **i** 行中值最小的元素, 且又是第 **j** 列中值最小的元素，则称之为该矩阵的一个马鞍点。请编程计算出 **m\*n** 的矩阵 **A** 的所有马鞍点。

如果某一行的最小值为 **2**，而这一行有 **3** 个 **2**，我们认为这三个 **2** 都是最小的元素。

**wtj** 觉得这个题目太简单了，于是给这个题目加了一个难度。

即现在给你 **n** 个相同大小的矩阵，每个矩阵的大小是 **x\*y**。如果某个位置，在每个矩阵都是马鞍点，我们称为套马点。

请你找出套马点的个数

## 输入：

第一行输入一个数 **n**，表示有 **n** 个矩阵  $1 < n < 10$

第二行输入两个数 **x, y** 表示矩阵有 **x** 行，每行 **y** 个数。 ( $1 < x, y < 10$ )

然后输入 **n\*x** 行，每行 **y** 个数。表示 **n** 个矩阵。

每个数的大小为 **[0,100]**

## 输出：

输出套马点的个数

## 样例输入：

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

## 样例输出：

```
1
```

## 提示

样例中

在第一个矩阵中 鞍点的位置为 (1,1)

在第二个矩阵中 鞍点的位置为 (1,1) (2,2)

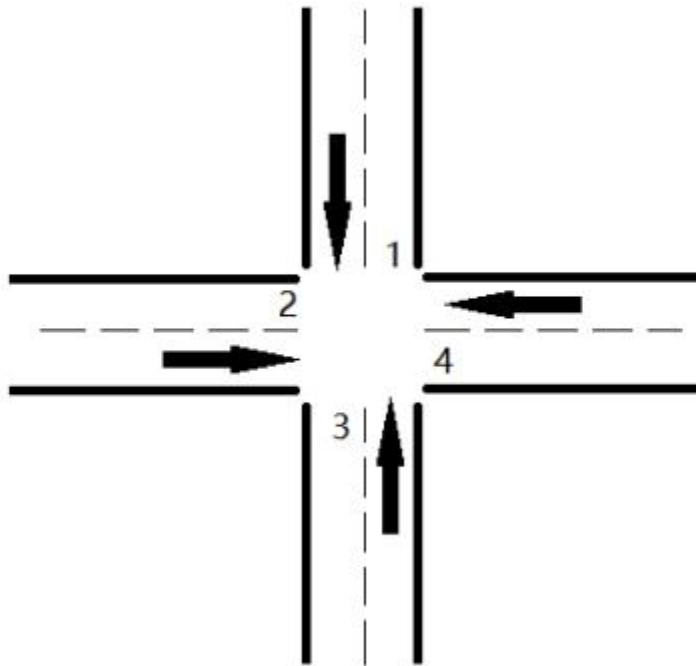
在第三个矩阵中 鞍点的位置为 (1,1) (2,4)

只有 (1,1) 这个位置在三个矩阵中都是鞍点，所以套马点的个数为 1

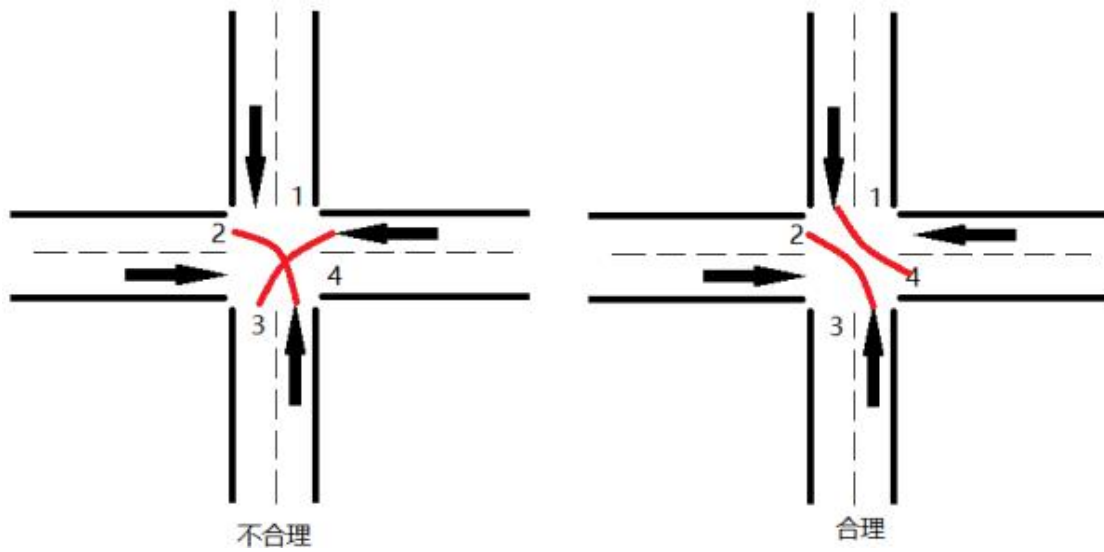
## 问题 H 繁忙的路口

### 描述：

无聊的会长今天出去打狂犬疫苗，因为被来福咬了。路上会长路过了一个十字路口，在等红绿灯，发现时间居然是 99 秒，会长觉得这很不合理，于是想要自己设计一套红绿灯算法，但是由于会长刚刚被喵咬了还没打狂犬疫苗，出现了短暂的神志不清，就连基本的红绿灯合理性都判断不出来了，你能帮帮会长吗？



如图，一个十字路口（双向道路），有四个信号灯（图中用数字标示），有两种信号左转和直行（右转可以直接走），只考虑红灯（停止信号）和绿灯（放行信号），没有黄灯。用“R”代表红灯，用“G”代表绿灯，从一号信号灯开始依次给出四个信号灯的信号（对于每个信号灯先输出左转信号再输出直行信号），请告诉会长该情况下是否合理，合理指没有两辆或以上的车辆通过路口的行车轨迹会相交。



**输入：**

一行八个字符代表四个信号灯的信号值

**输出：**

如果合理，请输出"perfect"，否则输出"terrible"（不带引号）。

**样例输入：**

RGRRRGRR

**样例输出：**

perfect

# 问题 I roll up

## 描述：

现在在进行一个，游戏中有 A,B,C,D,E,F 六个人在挑选卷心菜，卷心菜限量只有 120 个，每个卷心菜都是一样的，最后六个人中选的卷心菜的数量最多和最少的人（可能不止两个人，数量等于最大或最小值的人都要出局）都会出局，他们每个人都很聪明(A 除外，A 是工具人)，都会优先让自己不出局，再尽量让尽可能多的人出局，在此基础上希望自己的卷心菜最多，并且他们之间不能有任何交流，现在他们按 A,B,C,D,E,F 的顺序来挑选，每个人选的时间段不同且他们之间不会有交集，只能知道 120 个卷心菜还剩下多少个，120 个卷心菜可以不用都选完，但是至少保证每个人都有一个，也就是说第一个人不能直接选 116 个，至少要剩 5 个给后面的五个人。现在给出第一个人所选的卷心菜数量，请根据最优的策略，输出后面 B,C,D,E,F 会选择的卷心菜数量。

## 输入：

一个整数  $A(1 \leq A \leq 115)$  第一个人拿的卷心菜数量。

## 输出：

输出 5 个整数，分别代表 B,C,D,E,F 所选的卷心菜的数量。

## 样例输入：

60

## 样例输出：

56 1 1 1 1



# 问题 J 羊村快递站

## 描述：

马上就要到一年一度的"双十二"了，但是来自青青草原的懒羊羊却不想再次疯狂购物了，因为在"双十一"的时候懒羊羊被羊村快递站惩罚哭了，羊村快递站站长喜羊羊因为看到大家都及时来取快递，于是整出了如下的一套全新的取件机制：

- 1、所有新到达的快递都会进入待取通道，先到达的快递先进入待取通道，且位于待取通道口，每次只能在通道口取走一件快递，随后通道里的传送带开始转动让下一件快递位于通道口。
- 2、当要取的快递不位于待取通道口时，快递站管理员沸羊羊会将快递依次取出放进惩罚通道，直到取到取件羊的快递，最后被放进惩罚通道的位于惩罚通道口，先被放入的最后才能被取出。
- 3、当要取的快递位于惩罚通道口时，会直接取走自己的快递，随后整个通道就会向前移动让下一件快递位于通道口。
- 4、当要取的快递不位于惩罚通道口时，沸羊羊会将快递依次取出并放入快递车上，直到取到取件羊的快递，放入快递车的快递将会受到终极惩罚：被送回发货地，沸羊羊会将这些被打回的快递记录下来，当这些快递的取件羊来取件的时候，将会被告知取件失败。
- 5、每件进入惩罚通道的快递都会被记录下来，所以每个取件羊来到的快递站的时候，沸羊羊都会先告诉它要取的快递在哪个通道里。
- 6、每件快递的取件码和收件羊的取件码是对应的且是唯一的，取件羊会取走和自己取件码对应的快递。

聪明的喜羊羊觉得这套惩罚机制简直酷毙了，肯定能让大家及时来取快递。于是他想先行测试一下这套流程。给出快递的信息，输出每件快递被取走的次序。

## 输入：

第一行一个整数  $N$  代表快递个数。( $1 < N < 1e4$ )

第二行  $N$  个长度为四的取件码，代表每个快递的取件码（快件到达次序为输入顺序）。(0000-9999)

第三行一个整数  $Q$  代表取件人个数。

第四行  $Q$  个整数，代表每个人的取件码。

## 输出：

输出  $N$  个数，代表每个快递的取件次序（从 1 开始增加），如果取件失败则输出 -1，如果快件最终未被打回也未被取走则输出 0。

**样例输入：**

5  
0001 0010 0100 1000 1111  
4  
0001 1000 0010 0100

**样例输出：**

1 3 -1 2 0

# 问题 K 推箱子

## 描述：

想必大家对推箱子这个游戏都不陌生吧，推箱子是一个训练逻辑思考能力的游戏，在一个狭小的仓库中，要求把木箱放到指定的位置，稍不小心就会出现箱子无法移动或者通道被堵住的情况，所以需要巧妙的利用有限的空间和通道，合理安排移动的次序和位置，才能顺利的完成任务。小 y 经过大半个学期的努力，对 c 语言和代码能力都有了很大的提升，想写个破解推箱子游戏的代码练练手，即给定一个推箱子游戏地图，便可用最少的步数通过游戏，但这破解推箱子游戏的代码似乎对小 y 来说太过于简单(kunnan)，对于这么简单的代码就交给你这个萌新(dalao)了。

给定一个  $n$  行  $m$  列的地图，地图中只包含 0 到 5 中的数字：

- 0：空地
- 1：墙壁
- 2：箱子
- 3：人
- 4：箱子需要到达的指定位置
- 5：箱子已经在指定位置上

你需要操作地图上的人用最少的步数将箱子都放到指定位置上：

- w：往上走一步
- s：往下走一步
- a：往左走一步
- d：往右走一步

保证输入的地图是有解的，地图中只会存在 1 个人，箱子的数量和需要到达的指定位置数是相等的( $1 \leq \text{箱子数量} \leq 3$ )，地图的初始状态一定存在箱子没有在需要到达的位置上，人和箱子以及箱子需要到达的指定位置都被墙壁围在一个空间中。

## 输入：

第一行输入两个整数  $n, m$  ( $5 \leq n, m \leq 8$ )，表示地图的大小。

接下来输入  $n$  行，每行  $m$  个整数。对应第  $i$  行第  $j$  列的整数  $a[i][j]$ ，满足  $0 \leq a[i][j] \leq 5$ 。

## 输出：

第一行输出一个整数 **sum**，即将所有箱子放到指定位置上需要的最少步骤数。

第二行输出 **sum** 个字符，代表相应 **sum** 个步骤，如果最少步骤数有多种走法，则输出字典序最小的步骤。

样例输入：

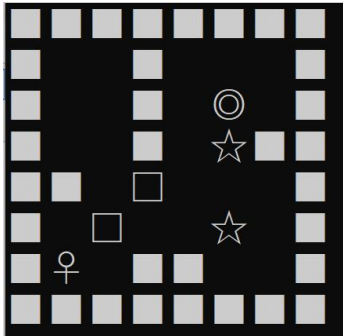
8 8  
1 1 1 1 1 1 1 1  
1 0 0 1 0 0 0 1  
1 0 0 1 0 5 0 1  
1 0 0 1 0 4 1 1  
1 1 0 2 0 0 0 1  
1 0 2 0 0 4 0 1  
1 3 0 1 1 0 0 1  
1 1 1 1 1 1 1 1

样例输出：

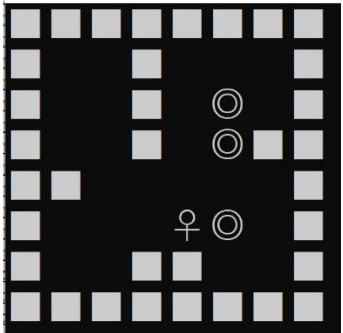
14  
wdwddsdwaaasdd

提示

推箱子规则：人不能通过墙壁，一次最多只能推动一个箱子。  
字典序：是基于字母顺序排列的单词按字母顺序排列的方法。  
样例初始时地图：



样例过关时的地图：



# 问题 L Alice and Bob

## 描述：

Alice 和 Bob 一起玩拿球游戏，每个球都有编号（1~n）不会重复，每一次最多只能拿 m 个球，至少拿一个球，最后谁不能拿了则失败。他们只能拿唯"1"球,唯"1"球是指编号和当前最大编号球编号只有唯一公约数的球。游戏遵守女士优先原则，每次都是 Alice 先拿。

## 输入：

输入一个数 T，表示他们进行了 T 次游戏( $1 \leq T \leq 1e5$ );

后面每行一组数据 n, m;( $1 \leq n \leq 1e7, 1 \leq m \leq 1e7$ )

## 输出：

胜利者的名字

## 样例输入：

```
4
2 3
2 1
8 1
8 2
```

## 样例输出：

```
Alice
Alice
Bob
Alice
```

# 问题 M 简单字符串匹配

## 描述：

JSU 某位少爷养了一群马，这世界上的马只有 A,B,C,D 四个等级，只有 B 能跑过 A，只有 C 能跑过 B，只有 D 能跑过 C，而自认为无敌的 D 碰到 A 时总会大意输给 A，因此只有 A 能跑过 D。现在知名贫困户也有一群马（肯定没有少爷多），现在贫困户向少爷发起赛马挑战，每次都是一匹对战一匹。现在双方都给出了马匹上场顺序，少爷的马很多从始至终都在上场，但是贫困户可以选择让自己先轮空一段时间，但是一旦上场就要连续对战，马匹上场顺序不可变且需要连续上场直至结束，例如少爷的马匹顺序为 ABCD，贫困户为 CD，那么贫困户可以等少爷的第二批马上场后再开始让自己的马上场，也就是如下匹配

某少爷：ABCD

贫困户：CD

现在贫困户想知道他最多有多少匹马能赢对方

## 输入：

第一行给出两个整数  $n, m$  ( $1 \leq m \leq n \leq 1e5$ )，分别为少爷的马匹数量和贫困户马匹数量。

第二行输入一个长度为  $n$  的字符串  $s1$ ，为少爷的马匹上场顺序。

第三行输入一个长度为  $m$  的字符串  $s2$ ，为贫困户的马匹上场顺序。

## 输出：

输出一个整数，贫困户最多能赢的马匹数

## 样例输入：

```
10 4
AAABBBBBBCD
CBBB
```

## 样例输出：

```
2
```