

# 多校联合训练第六场

## 一、题目概览

中文题目名称	0的个数	吃月饼	黑洞2	遍历
英文题目名称	zero	moon	blackhole	dfs
可执行文件名	zero	moon	blackhole	dfs
输入文件名	zero.in	moon.in	blackhole.in	dfs.in
输出文件名	zero.out	moon.out	blackhole.out	dfs.out
时间限制	1秒	1.0秒	1秒	1.0秒
空间限制	512MB	512MB	512MB	512MB
测试点数目	20	10	10	10
测试点分值	5	10	10	10
题目类型	传统	传统	传统	传统
比较方式	全文比较	全文比较	全文比较	全文比较
是否有部分分	是	是	是	是

## 二、注意事项：

- 1.文件名（程序名和输入输出文件名）必须使用小写。
- 2.C/C++中函数main()的返回值类型必须是int，程序正常结束时的返回值必须是0。
- 3.评测环境为Linux，使用lemonlime进行评测。
- 4.选手不得使用SSH等命令。
- 5.选手不得使用内嵌汇编，#pragma等指令。
- 6.评测时使用 C++14 环境，同时开启 O2 优化，栈空间和内存限制相同。

## 0的个数(zero)

### 【题目描述】

小明喜欢玩数字游戏。

小明选了  $n$  个数，他选择用乘法运算将他们乘在一起，现在小明想知道这个数末尾有多少个 0。

## 【输入格式】

第一行一个整数  $n$ ，表示如上。

第二行  $n$  个整数，表示需要用乘法运算乘起来的数

## 【输出格式】

一个整数，表示末尾0的数量

## 【样例 1 输入】

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

## 【样例 1 输出】

```
1
```

样例解释：  $1 \times 2 \times 3 \times 4 \times 5 = 120$ ，有 1 个 0

## 【样例 2】

见下发文件

## 【子任务】

对于 70% 的数据，  $1 \leq n \leq 10, 1 \leq a_i \leq 10$ 。

对于 90% 的数据，  $1 \leq n \leq 10^3, 1 \leq a_i \leq 10^3$

对于100%的数据，  $1 \leq n \leq 10^6, 1 \leq a_i \leq 10^9$ 。

# 吃月饼 (moon)

## 【题目描述】

小明和  $n - 1$  个人坐在一起，他们决定分月饼，但是月饼只有  $n - 1$  个

为了公平，他们决定用经典的约瑟夫环方式，围坐成一圈，由 1 号位开始报数，报到  $m$  的那个人可以吃一个月饼，他的下一个人又从 1 开始报数，数到  $m$  的人再次吃月饼；...直到留下一个人。

小明不喜欢吃月饼，他偷偷靠电话找到了你，希望你可以找到不用吃月饼的座位。

## 【输入格式】

第一行两个整数  $n, m$ 。

## 【输出格式】

输出一个整数，表示答案。

## 【样例 1 输入】

6 2

## 【样例 1 输出】

5

## 【样例 2】

见下发文件

## 【子任务】

对于30%的数据,  $1 \leq n, m \leq 1000$ 。

对于50%的数据,  $1 \leq n \leq 1000, 1 \leq m \leq 10^6$ 。

对于100%的数据,  $1 \leq n, m \leq 10^6$ 。

# 黑洞2 (blackhole)

## 【题目描述】

小明有一个有  $n$  行  $m$  列的棋盘, 一共有  $n \times m$  个棋盘格子。

有一天他的棋盘被高维度生物入侵了, 造出了若干个黑洞, 第  $i$  个黑洞位于棋盘的第  $x_i$  行, 第  $y_i$  列, 而其他格子都是正常的, 每个格子上都放有一颗棋子。小明想到了一个有趣的游戏, 他可以通过倾斜棋盘使得棋子发生一模一样的位移。具体来说, 小明可以通过倾斜棋盘, 使得棋盘上所有棋子同时往上下左右位移一个格子。

举个例子, 当前棋子位于  $(i, j)$ , 小明设计了倾斜棋盘的按钮。

1. 按钮 **U**: 它将移动到  $(i - 1, j)$
2. 按钮 **D**: 它将移动到  $(i + 1, j)$
3. 按钮 **L**: 它将移动到  $(i, j - 1)$
4. 按钮 **R**: 它将移动到  $(i, j + 1)$

如果一颗棋子掉进黑洞里或掉出棋盘, 那么该棋子将离开棋盘不再回来。

现在小明想知道, 哪些棋子能成为最后的天选之棋子, 即通过若干次操作后只有它仍然在棋盘上。

## 【输入格式】

多组数据。

第一行一个整数  $T$ , 表示数据组数。

对于每组数据, 第一行两个整数  $n, m$ , 表示棋盘大小。

接下来  $n$  行, 每行一个长度为  $m$  的字符串, 表示棋盘第  $i$  行的状态, **.** 表示正常格子, **o** 表示黑洞。

## 【输出格式】

对于每组数据输出一行一个整数，表示能成为最后留下的棋子的数量。

## 【样例 1 输入】

```
3
2 5
.00..
0..0.
1 3
0.0
1 3
.0.
```

## 【样例 1 输出】

```
3
1
0
```

样例解释：对于第一组数据，(1,4)可以通过 **RD** 成为留在棋盘上唯一一颗棋子，(1,5)可以通过 **D** 成为留在棋盘上唯一一颗棋子，(2,5) 可以通过 **U** 成为留在棋盘上唯一一颗棋子。一共3种可能。

## 【样例 2】

见下发文件

## 【子任务】

对于 30% 的数据， $1 \leq n, m \leq 4$ 。

对于 70% 的数据， $1 \leq n, m \leq 15$ 。

对于100%的数据， $1 \leq T \leq 5, 1 \leq n, m \leq 40$ 。

# 遍历 (dfs)

## 【题目描述】

007被困在了一个树形迷宫中。

他此时处在的点，在他的认知中，是这棵树的根节点。

007有一种特殊的能力，他不会重复探索同一个房间，并且每次他都会选择最左边第一个没有探索过的房间进行探索，如果没有，他就退回上一个房间，最后他会回到最初的房间

每个房间都有标记，但是标记可能重复，007将这些标记按顺序记录了下来。

现在007已经有了这个标记序列，他想知道，这个迷宫的形状有多少种可能（答案对 $1e9+7$ 取模）

## 【输入格式】

一个只包含小写字母的字符串，表示007得到的序列

## 【输出格式】

一个整数，表示答案

## 【样例 1 输入】

```
ababa
```

## 【样例 1 输出】

```
2
```

## 【样例 2】

见下发文件

## 【数据范围】

令 $n$ 表示字符串长度

对于30%的数据， $1 \leq n \leq 10$ 。

对于100%的数据， $1 \leq n \leq 300$ 。