

# 擂台赛

时间限制：1.0 秒

刷新 ↺

空间限制：512 MiB

## 题目描述

有  $n$  个选手在打擂台赛，编号为 1 到  $n$ ，保证  $n$  是 2 的幂次且  $n \leq 8$ 。

一开始会将  $n$  个选手分成  $n/2$  组，每组 2 个选手。组内比赛之后胜出者进入下一轮，直到剩下最后一位选手为最终的胜者。

当  $n = 4$  时，先将 4 个选手分成 2 组，第一组的胜者和第二组的胜者进行最后的比赛。

当  $n = 8$  时，先将 8 个选手分成 4 组，第一组的胜者和第二组的胜者进行下一轮的比赛，第三组的胜者和第四组的胜者进行下一轮的比赛，两边的胜者再进行最后的比赛。

给出  $n$  个选手之间互相比赛的胜负情况，问有多少种分组的方案使得最后编号为 1 的选手胜出。两种方案被认为是不同的，当且仅当两种方案中存在至少一位选手分到的组的编号不同。

## 输入格式

从标准输入读入数据。

输入的第一行包含两个正整数  $n$ ，表示参加擂台赛的人数。

接下来  $n$  行，每行  $n$  个数字。其中第  $i$  行，第  $j$  个数字  $a_{i,j}$  表示编号为  $i$  的选手和编号为  $j$  的选手比赛的胜负情况， $a_{i,j} = 1$  表示编号为  $i$  的选手胜出， $a_{i,j} = -1$  表示编号为  $j$  的选手胜出。当  $i = j$  时， $a_{i,j} = 0$ 。

## 输出格式

输出到标准输出。

输出一行包含一个非负整数表示对应的方案数。

# 样例1输入

```
4
0 1 1 1
-1 0 1 1
-1 -1 0 1
-1 -1 -1 0
```

# 样例1输出

```
6
```

# 样例1解释

一共有六种方案， 以下是对应分组的情况：

- (1,2) (3,4)
- (1,3) (2,4)
- (1,4) (2,3)
- (3,4) (1,2)
- (2,4) (1,3)
- (2,3) (1,4)

# 数据范围

本题共 10 个测试点， 每个测试点 10 分。

对于所有测试数据， 满足  $2 \leq n \leq 8, -1 \leq a_{i,j} \leq 1$ 。

对于编号为 1 ~ 2 的测试点， 满足  $n = 2$ 。

对于编号为 3 ~ 6 的测试点， 满足  $n = 4$ 。

对于编号为 7 ~ 10 的测试点， 满足  $n = 8$ 。

## 语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	g++ with std=c++14	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B
1	java	javac		65536 B
2	python3	python3		65536 B

递交历史

#	状态	时间
---	----	----

221188	Accepted	2024-03-25 14:18:16
--------	----------	---------------------

1

# 栈

时间限制：1.0 秒

刷新 ↻

空间限制：512 MiB

## 题目描述

给定  $n$  个初始为空的栈，你需要维护  $m$  个操作，每个操作为以下三种之一：

- $1\ x\ w\ c$ ：在第  $x$  个栈中加入  $c$  个  $w$ ，你需要回答加入后第  $x$  个栈内的所有数之和；
- $2\ x\ c$ ：在第  $x$  个栈中弹出末尾  $c$  个数（保证第  $x$  个栈内有至少  $c$  个数），你需要回答弹出的  $c$  个数之和；
- $3\ x\ y$ ：依次将第  $x$  个栈的数弹出并加入到第  $y$  个栈，你需要回答加入后第  $y$  个栈内的所有数之和。

## 输入格式

从标准输入读入数据。

输入的第一行包含两个正整数  $n, m$ ，分别表示栈的个数和需要执行的操作个数。

接下来  $m$  行，每行按上述格式描述一个操作。

## 输出格式

输出到标准输出。

输出  $m$  行，每行一个非负整数表示对应操作应回答的结果。

## 样例1输入

3 7  
1 1 3 2  
1 2 2 3  
1 2 4 1  
3 2 3  
2 3 2  
3 3 1  
2 1 4

## 样例1输出

6  
6  
10  
10  
4  
12  
12

## 样例1解释

- 第 1 次操作后，第 1 个栈变为 3, 3，答案为  $3 + 3 = 6$ ；
- 第 2 次操作后，第 2 个栈变为 2, 2, 2，答案为  $2 + 2 + 2 = 6$ ；
- 第 3 次操作后，第 2 个栈变为 2, 2, 2, 4，答案为  $2 + 2 + 2 + 4 = 10$ ；
- 第 4 次操作依次弹出 4, 2, 2, 2，操作后第 2 个栈为空，第 3 个栈变为 4, 2, 2, 2，答案为  $4 + 2 + 2 + 2 = 10$ ；
- 第 5 次操作依次弹出 2, 2，操作后第 3 个栈变为 4, 2，答案为  $2 + 2 = 4$ ；
- 第 6 次操作依次弹出 2, 4，操作后第 3 个栈为空，第 1 个栈变为 3, 3, 2, 4，答案为  $3 + 3 + 2 + 4 = 12$ ；
- 第 7 次操作依次弹出 4, 2, 3, 3，操作后第 1 个栈为空，答案为  $4 + 2 + 3 + 3 = 12$ 。

## 数据范围

对于所有测试数据，满足  $1 \leq n, m, w \leq 2 \times 10^5$ ， $1 \leq x, y \leq n$ ， $x \neq y$ ， $1 \leq c \leq 10^8$ 。

测试点编号	$n, m, w \leq$	$c \leq$
1 ~ 2	2000	1
3 ~ 4	2000	$10^8$
5 ~ 6	$10^5$	1

7 ~ 8

$10^5$

$10^8$

9 ~ 10

$2 \times 10^5$

$10^8$

语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	g++ with std=c++14	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B
1	java	javac		65536 B
2	python3	python3		65536 B


递交历史

#	状态	时间
222014	Wrong Answer	2024-03-25 16:23:38
221895	Wrong Answer	2024-03-25 16:09:43
221877	Wrong Answer	2024-03-25 16:06:20

1

# 指针

时间限制：1.0 秒

刷新 

空间限制：512 MiB

## 题目描述

有  $10^9$  台设备分布在一条数轴上，第  $i$  台设备的坐标为  $i$ 。有  $n$  位维修工，初始时第  $i$  位维修工的位置为  $a_i$ 。

这些设备共发生了  $m$  次故障，第  $j$  次故障的设备为  $b_j$ ，你需要指定某一名维修工维修设备  $b_j$ ，他将从他当前所在的位置移动到位置  $b_j$ 。维修工从位置  $x$  移动到位置  $y$  需要花费  $|x - y|$  的代价。

你需要合理调配维修工，在每次故障发生后及时完成维修，即必须依次完成  $m$  次维修。求所有维修的代价总和的最小值。

## 输入格式

从标准输入读入数据。

输入的第一行包含两个正整数  $n, m$ ，分别表示维修工个数和故障次数。

输入的第二行包含  $n$  个正整数  $a_1, a_2, \dots, a_n$ ，分别表示每个维修工的初始位置。

输入的第三行包含  $m$  个正整数  $b_1, b_2, \dots, b_m$ ，分别表示每次故障的设备。

## 输出格式

输出到标准输出。

输出一行一个非负整数表示代价总和的最小值。

## 样例1输入

2 5  
3 6  
4 8 1 5 7

# 样例1输出

11

# 样例1解释

- 第 1 次维修，维修工 1 移动到位置 4，代价为  $|3 - 4| = 1$ ；
  - 第 2 次维修，维修工 2 移动到位置 8，代价为  $|6 - 8| = 2$ ；
  - 第 3 次维修，维修工 1 移动到位置 1，代价为  $|4 - 1| = 3$ ；
  - 第 4 次维修，维修工 1 移动到位置 5，代价为  $|1 - 5| = 4$ ；
  - 第 5 次维修，维修工 2 移动到位置 7，代价为  $|8 - 7| = 1$ ；
- 代价总和为  $1 + 2 + 3 + 4 + 1 = 11$ 。

# 数据范围

对于所有测试数据，满足  $1 \leq n, m \leq 600$ ,  $1 \leq a_i, b_j \leq 10^9$ 。

测试点编号	$n \leq$	$m \leq$	$a_i, b_j \leq$
1 ~ 2	10	6	$10^5$
3 ~ 4	2	600	$10^5$
5 ~ 6	3	600	$10^5$
7 ~ 8	5	100	10
9 ~ 13	200	200	$10^5$
14 ~ 20	600	600	$10^9$

## 语言和编译选项

#	名称	编译器	额外参数	代码长度限制
0	g++ with std=c++14	g++	-O2 -std=c++14 -DONLINE_JUDGE	65536 B



1	java	javac	65536 B
2	python3	python3	65536 B

递交历史			
#	状态	时间	
222357	Wrong Answer	2024-03-25 17:04:51	
222253	Time Limit Exceeded	2024-03-25 16:52:22	1