

# 1024 Coding Challenge

## Algorithmic Track

2025 Programmer's Day

本场比赛将在 2025/10/24 UTC<sup>+8</sup> 20 : 00 举行，为期 48 hr。

题号	题目名称	题目类型	时间限制	内存限制	测试点数目
A	台风	传统题	1.0 秒	512 MiB	5
B	Run-length Encoding	传统题	1.0 秒	512 MiB	25
C	神秘的三角形	传统题	1.0 秒	512 MiB	10
D	图书整理计划	传统题	1.0 秒	512 MiB	5
E	Fetch-Decode-Execute	传统题	1.0 秒	512 MiB	10
F	不神秘构造题	Special Judge	1.0 秒	512 MiB	10
G	星辉廊道	传统题	1.0 秒	512 MiB	20
H	File Restoration	传统题	1.0 秒	512 MiB	20
I	扫雷	传统题	5.0 秒	512 MiB	20
J	1024 <sub>1024</sub> Puzzle	传统题	1.0 秒	512 MiB	25
K	抽卡打怪	传统题	1.0 秒	512 MiB	10
L	「星光渐明之时」	传统题	1.0 秒	512 MiB	25

### 注意事项（请仔细阅读）：

- 本比赛支持的语言有：Pseudocode, Python3, C, C++, Java, Javascript, 和 Go 语言。其中 C++ 语言开启 O2 优化。
- 对比赛内容有疑问的，可以私下私聊 Staff 询问。请不要在比赛群聊公开讨论题目。
- 程序可使用的栈空间限制与题目的内存限制一致。
- 源程序中不允许包含攻击本次竞赛的侮辱性不文明语言，也不允许出现存在不当，损害编译器的代码。
- 严禁任何形式的作弊，包括但不限于 AI 生成工具（如 ChatGPT 等），或是和其他参赛者私下交换答案。
- 严禁对竞赛平台进行任何形式的攻略攻击。已实施防火墙以保护网站。如果你被阻止，请尝试关闭你的 VPN。
- 违反比赛规则的人员将会被取消参赛资格。
- This PDF is not a puzzle.
- 那么，祝你们好运！

## [A] 台风

### 【题目描述】

据深圳国际交流书院气象台显示，“超强台风”【数据删除】”已成为今年以来全球风王，将于今日深圳沿海登陆，预计登陆强度最低也将是强台风级！

不少深国交的学生为了他们的生命安全，向学校申请今天停课！

请你模拟学校 PSO，为申请停课的学生们回复邮件。

### 【输入格式】

一行一个字符串，表示该学生申请停课的理由。

### 【输出格式】

一行一个字符串 Reject! 表示学校的回复。

### 【样例】

#### 样例 1

In

I hope PSO will be suspending our hybrid learning due to current weather condition!

Out

Reject!

### 【数据范围】

数据保证输入的字符串不存在违法和不道德要素，包括但不限于辱骂语言等。

## [B] Run-length Encoding

### 【题目描述】

Run-length Encoding 是一种比较简单的压缩算法，其基本思想是将连续的几个相同字母写成字母加上出现次数的形式。比如说 LLLLLLLLLL 会被写成 L13，也就是 13 个字符 L。

举个例子，一个字符串 SSSCCIEEEE 会被编码成为 S3C2I1E4。

已知初始字符串，请通过 Run-length Encoding 算法将原来的字符串压缩至密文。

### 【输入格式】

一个字符串，表示初始字符串。

### 【输出格式】

一个字符串，表示加密过后的字符串。

### 【样例】

#### 样例 1

In

AAPPPPLLEEECCCCAAKKSN

Out

A2P4L3E4C4A2K2S1N1

### 【数据范围】

字符串长度小于  $10^5$ .

## [C] 神秘的三角形

### 【题目描述】

考古学家在一次探险中发现了一些神秘石板，每块石板上都刻着一个三角形的相关数据。研究人员希望通过这些数据来计算出三角形的实际面积，从而推测古代工匠的设计意图。

石板上的数据有两种形式：

1. 三角形的三条边长  $a, b, c$ 。
2. 三角形的底和高  $base, height$ 。

请你编写一个算法，根据输入类型，计算并输出这些三角形的面积。

### 【输入格式】

第一行：一个整数  $T$ ，表示测试用例数量。

对于每组测试：

- 首先输入一个整数  $t$ （表示输入类型）。
- 若  $t = 1$ ，接下来输入三个整数  $a, b, c$ ，表示三角形的三条边长。
- 若  $t = 2$ ，接下来输入两个整数  $base, height$ ，表示三角形的底和高。

### 【输出格式】

对于每组测试，输出一个浮点数，表示三角形的面积，保留两位小数。

### 【样例】

#### 样例 1

##### In

```
2
1 3 4 5
2 10 6
```

##### Out

```
6.00
30.00
```

### 【数据范围】

- $1 \leq T \leq 10^5$

- $1 \leq a, b, c, \text{base}, \text{height} \leq 1000$
- 当输入为三条边长时, 请使用 海伦公式 (Heron's Formula):

$$s = \frac{a + b + c}{2}, \quad \text{area} = \sqrt{s(s - a)(s - b)(s - c)}$$

在 pseudocode 中, 可以使用 `SQRT(X:REAL) -> REAL` 来计算平方根

- 当输入为底和高时, 使用公式:

$$\text{area} = \frac{1}{2} \times \text{base} \times \text{height}$$

## [D] 图书馆整理计划

### 【题目描述】

SCIE 图书馆计划在接下来的  $N$  天内整理图书。第  $i$  天需要处理  $a_i$  本书。整理流程如下：

1. 汇总：把所有  $N$  天要处理的书合并在一起，不区分天数。
2. 装箱：每装满一个纸箱恰好需要  $B$  本书；每箱处理（装箱+搬运）耗时  $X$  分钟。
3. 单本处理：无法装满一箱的剩余书籍，每本单独处理，耗时  $Y$  分钟/本。

请你帮助图书馆管理员计算完成整个整理计划所需的总时间！

### 【输入格式】

第一行四个整数： $N, B, X, Y$ 。

第二行  $N$  个整数： $a_1, a_2, \dots, a_N$ 。

### 【输出格式】

一行一个整数，代表总耗时（分钟）。

### 【样例】

#### 样例 1

In

```
3 10 15 2
4 11 5
```

Out

```
30
```

### 【数据范围】

- $1 \leq N \leq 10^5$
- $0 \leq a_i \leq 10^6$
- $1 \leq B \leq 10^6$
- $0 \leq X, Y \leq 10^6$ .

## [E] Fetch-Decode-Execute

### 【题目描述】

现代计算机的 CPU 执行程序时，会不断重复 Fetch-Decode-Execute 的过程循环。本题要求你模拟一个计算机，处理 FDE 循环。

### CPU 结构

- 一个 32 位带符号累加寄存器 (Accumulator) A，初始值为 0。
- 一个 32 位带符号程序计数器 (Program Counter) PC，初始值为 0，指向当前指令编号。
- 指令存储器 M，共存放 N 条指令（编号 0 … N-1）。
- 数据存储器 D，大小为 256 个整数单元（编号 0..255），初始值全为 0。
- 一个循环计数器，用来统计 CPU 执行了多少个 FDE 周期。

### 指令集

每条指令占用一条指令存储单元：

1. LOAD x —— A = D[x]
2. STORE x —— D[x] = A
3. ADD x —— A = A + D[x]
4. SUB x —— A = A - D[x]
5. MUL x —— A = A \* D[x]
6. MOV v —— A = v (更改累加器的值为 v)
7. JMP i —— PC = i
8. JZ i —— 若 A == 0，则 PC = i；否则 PC = PC + 1
9. JNZ i —— 若 A != 0，则 PC = i；否则 PC = PC + 1
10. NOP —— 空操作（什么都不做）
11. HALT —— 程序终止（但该指令仍算一个周期）

### 执行规则

- FDE 循环过程
  - Fetch: 取出 PC 指向的指令
  - Decode and Execute: 修改 A、D[...] 或 PC
  - PC Update: 若指令没有显式修改 PC（如 JMP、JZ、JNZ），则执行后 PC = PC + 1
- 若 PC 变为负数或超出 [0, N-1]，则程序立刻崩溃，判定为终止。
- 若循环次数超过  $10^7$ ，判定为死循环，程序强制停止。
- 所有算术运算均在 32 位带符号整数范围内进行（补码表示），需要考虑溢出。

## 【输入格式】

第一行：整数  $N$  ( $1 \leq N \leq 10^5$ )，表示指令条数。

接下来  $N$  行：每行一条指令。

- 指令与参数之间用一个空格分隔。
- $x$  和  $i$  是整数， $v$  是可能为负数的整数。
- 对于数据内存操作，保证  $0 \leq x \leq 255$ 。
- 跳转目标  $i$  可以为任意整数（若超出范围则会崩溃）。

## 【输出格式】

程序终止后（由 HALT、崩溃或超时）：输出三行

- 一个单词表示终止原因：HALT（正常结束）、CRASH（越界或非法）、TIMEOUT（循环超时）。
- 累加器 A 的最终值（32 位带符号整数）。
- 执行的总周期数。

## 【样例】

### 样例 1

#### In

```
7
MOV 5
STORE 0
MOV 10
STORE 1
LOAD 0
ADD 1
HALT
```

#### Out

```
HALT
15
7
```

### 样例解释 #1

- MOV 5 → A=5
- STORE 0 → D[0]=5

- MOV 10 → A=10
- STORE 1 → D[1]=10
- LOAD 0 → A=5
- ADD 1 → A=15
- HALT → 停止

共 7 个周期，最终 A=15。

## [F] 不神秘构造题

### 【题目描述】

你有一个  $3 \times 3$  的正方形，如图所示，标着从 1 到 9 的数字。这是一个  $3 \times 3$  网格的初始状态。

1	2	3
4	5	6
7	8	9

你可以对这个网格进行若干次操作。对于每次操作，选择一个  $2 \times 2$  区域，将区域内的数字顺时针旋转一格。

如图所示，总共有四种可能的操作，我们分别叫做 A/B/C/D。

	A		B																		
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	->	<table border="1"><tr><td>4</td><td>1</td><td>3</td></tr><tr><td>5</td><td>2</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	4	1	3	5	2	6	7	8	9	
1	2	3																			
4	5	6																			
7	8	9																			
4	1	3																			
5	2	6																			
7	8	9																			
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	->	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	
1	2	3																			
4	5	6																			
7	8	9																			
1	2	3																			
4	5	6																			
7	8	9																			
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	C	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>7</td><td>4</td><td>6</td></tr><tr><td>8</td><td>5</td><td>9</td></tr></table>	1	2	3	7	4	6	8	5	9	D
1	2	3																			
4	5	6																			
7	8	9																			
1	2	3																			
7	4	6																			
8	5	9																			
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	->	<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9	->
1	2	3																			
4	5	6																			
7	8	9																			
1	2	3																			
4	5	6																			
7	8	9																			
<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>5</td><td>6</td></tr><tr><td>7</td><td>8</td><td>9</td></tr></table>	1	2	3	4	5	6	7	8	9		<table border="1"><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>4</td><td>8</td><td>5</td></tr><tr><td>7</td><td>9</td><td>6</td></tr></table>	1	2	3	4	8	5	7	9	6	
1	2	3																			
4	5	6																			
7	8	9																			
1	2	3																			
4	8	5																			
7	9	6																			

给定一个目标状态。请你判断是否构造一系列操作可以将初始状态转变成目标状态，并构造一种操作方案。

### 【输入格式】

共 3 行，每行 3 个数字，表示目标状态。

### 【输出格式】

第一行输出 YES 或者 NO，代表是否可以把初始状态转变成目标状态。

若你输出了 YES，第二行一个长度小于  $10^3$  的由  $ABCD$  四个字母中构成的字符串，表示你构造的操作方案。

## 【样例】

### 样例 1

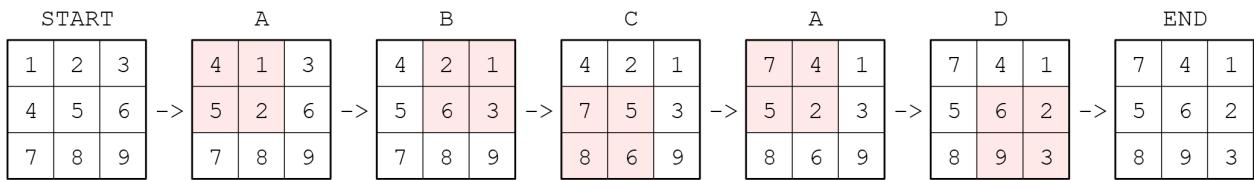
#### In

```
7 4 1
5 6 2
8 9 3
```

#### Out

```
YES
ABCAD
```

### 样例解释 #1



## [G] 星辉廊道

### 【题目描述】

奥拉星门即将关闭，唯有沿星辉廊道从起点港（港口 1）走到终点港（港口  $N$ ）才能撤离。廊道由若干段光桥组成，每段光桥拥有一个「调和度」 $w$ 。

连续经过多座光桥后，整体调和度会逐级衰减为所有段调和度的「共抑值」（即按位与）。你的任务是在不重复经过任何港口的前提下，使共抑值最大。

### 【输入格式】

第一行两个整数  $N, M$ 。

随后  $M$  行，每行三个整数  $u, v, w$ ，表示一条双向光桥连接港口  $u$  与  $v$ ，调和度为  $w$ 。

### 【输出格式】

输出一个整数，表示最大共抑值。

### 【样例】

#### 样例 1

In

```
3 3
1 2 5
2 3 3
1 3 6
```

Out

```
6
```

#### 样例 2

In

```
5 6
1 2 679161504
2 3 434058576
3 4 1009289799
2 5 602565249
1 4 59966696
3 1 927242445
```

**Out**

543828608

**【数据范围】**

- $2 \leq N \leq 2 \times 10^5$
- $N - 1 \leq M \leq 2 \times 10^5$
- $1 \leq u, v \leq N$
- $0 \leq w \leq 2^{30}$
- 所有港口互相连通。

分值	数据范围
10	$N \leq 200$
20	$N \leq 1000, M \leq 3000$
30	$N \leq 2 \times 10^4$
40	无额外限制

## [H] File Restoration

### 【题目描述】

你正在维护一个由  $n$  个文件夹组成的文件系统，编号从 1 到  $n$ 。文件夹 1 是根目录。

每个文件夹都有一个状态，可能是正常的（状态为 1）或损坏的（状态为 0）。

你的目标是通过执行一系列操作，使得所有文件夹的状态都变为正常（状态为 1）。

你可以执行以下两种操作：

1. 选择一个文件夹  $v$ ，将其状态翻转（即从 0 变为 1，或从 1 变为 0）。
2. 选择一个文件夹  $v$ ，将以  $v$  为根的子树中的所有文件夹（包括  $v$  本身）的状态全部翻转。

### 【输入格式】

第一行包含一个整数  $n$ ，表示文件夹的数量。

接下来  $n - 1$  行，每行包含两个整数  $u$  和  $v$ ，表示文件夹  $u$  和文件夹  $v$  之间存在父子关系（注意：整个结构是一棵树，文件夹 1 是根目录）。

接下来一行包含  $n$  个整数，表示每个文件夹的初始状态（0 表示损坏，1 表示正常）。

### 【输出格式】

输出一个整数，表示最少需要执行多少次操作才能使所有文件夹的状态都变为正常（状态为 1）。

### 【样例】

#### 样例 1

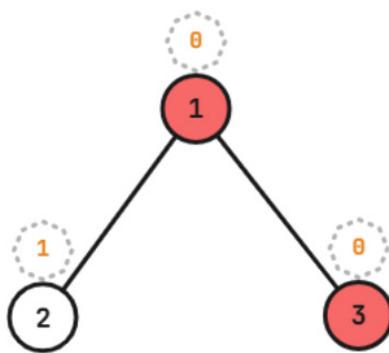
##### In

```
3
0 1 0
1 2
1 3
```

##### Out

```
2
```

## 样例解释 # 1



仅需将标红的文件夹进行操作 1 即可。

### 【数据范围】

对于所有数据有  $1 \leq n \leq 10^5$ 。

无部分分设置。

## [I] 扫雷

### 【题目描述】

在一个  $n \times n$  的扫雷地图中，初始没有地雷，你需要在下面进行 4 种操作。

- 将第  $x$  行，第  $y$  列的地方加入一颗地雷。
- 删除第  $x$  行，第  $y$  列的地方的地雷。
- 将整个地图除了第  $x$  行，第  $y$  列这一个地雷以外的所有地雷全部反转（即原先没有地雷的地方会变成地雷，而有地雷的地方地雷会消失）。
- 求出整个扫雷地图中所有数字的总和（扫雷方格中数字表示其周围的 8 个方格隐藏了几颗雷）。

### 【输入格式】

第一行两个整数  $n, m$ ，分别表示扫雷地图的大小和操作的总个数。

接下来的  $m$  行每行包含一些整数，表示一个操作，具体如下：

- 1  $x$   $y$ : 将第  $x$  行，第  $y$  列的地方加入一颗地雷。
- 2  $x$   $y$ : 删除第  $x$  行，第  $y$  列的地方的地雷；若该地方原本就没有地雷则无效。
- 3  $x$   $y$ : 将整个地图除了第  $x$  行，第  $y$  列这一个地雷以外的所有地雷全部反转。
- 4: 询问整个扫雷地图中所有数字的总和。

### 【输出格式】

若干行整数，表示所有操作 4 的结果。

### 【样例】

#### 样例 1

In

```
5 6
1 4 2
1 3 3
4
3 3 2
2 3 3
4
```

**Out**

14

18

**样例解释 # 1**

	1	2	3	4	5
1	0	0	0	0	0
2	0	1	1	1	0
3	1	1	X	1	0
4	1	X	1	1	0
5	1	1	1	0	0

ans=14

	1	2	3	4	5
1	X	X	X	X	X
2	X	X	X	X	x
3	X	6	6	X	X
4	X	6	X	X	X
5	X	X	X	X	X

ans=18

**【数据范围】**

- $1 \leq n \leq 10^6$
- $1 \leq m \leq 10^5$ .

- 特殊性质 A: 保证不出现操作 3。

分值	数据范围	性质
20	$1 \leq n \leq 100, 1 \leq m \leq 100$	/
20	$1 \leq n \leq 10^3, 1 \leq m \leq 10^3$	/
10	$1 \leq n \leq 10^6, 1 \leq m \leq 10^6$	<b>A</b>
50	无额外限制	/

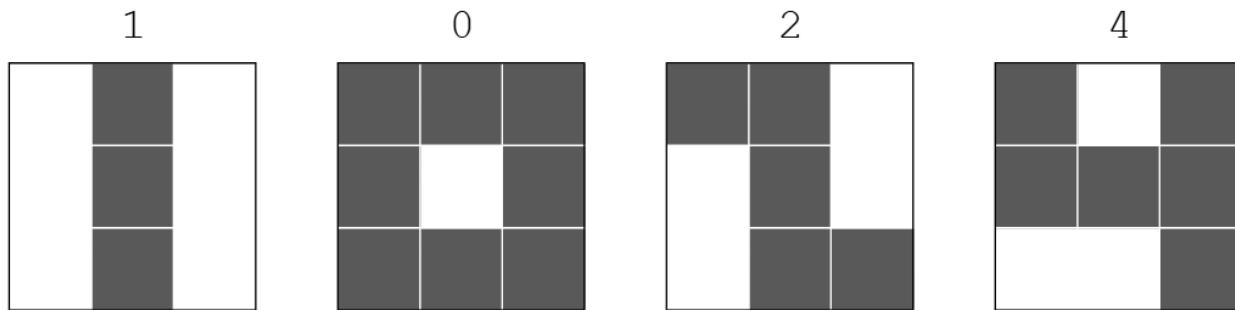
## [J] 1024<sub>1024</sub> Puzzle

### 【题目描述】

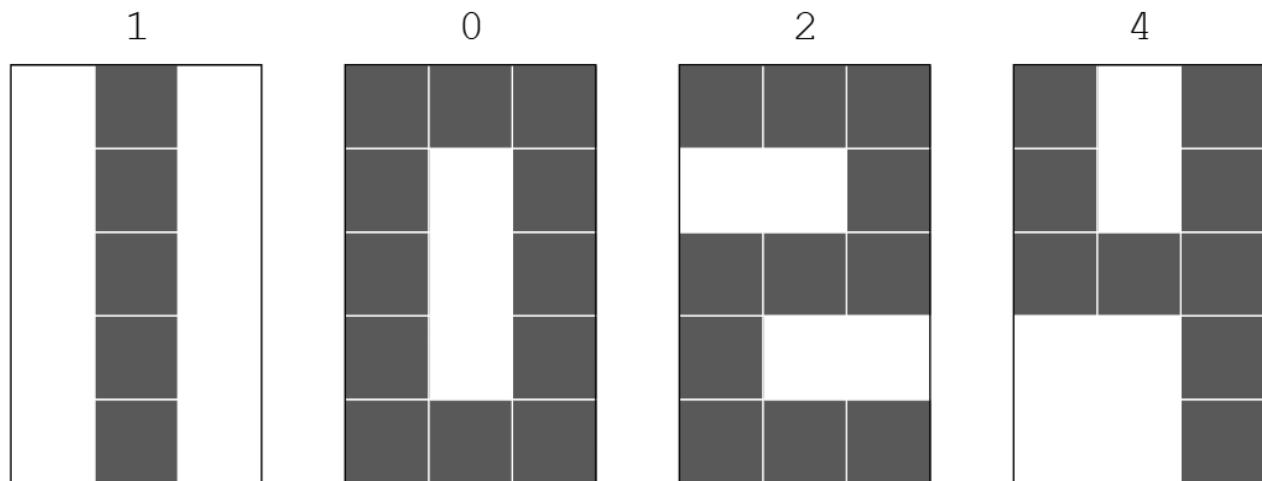
Yin 和 Yang 在玩游戏。Yin 先行。

游戏盘面是一个  $m$  行 3 列的方格中。

Yin 可以使用这些图形：



Yang 可以使用这些图形：



请注意，“1” 必须放在中间。

轮到某一方的时候，他需要选择一个自己可以使用的图形，从最上方放下去。这个图形将往下坠落直到接触到地板或之前放过的图形。

当某一方放下任何图形图形都会超出边界时，此时他不用放图形，游戏结算，分数为二人放下的图形的对应数字和。

再注意，如果有可以放下的图形则必须放。

Yin 想要让分数最大，Yang 想要让分数最小。最优策略下求最后的分数。

### 【输入格式】

本题一个测试点有多组数据。

第一行一个正整数  $T$  代表数据组数。

对于每组数据，一行一个正整数  $m$ 。

## 【输出格式】

对于每组数据，输出一行一个数代表最终分数。

## 【样例】

### 样例 1

#### In

```
4
2
4
6
8
```

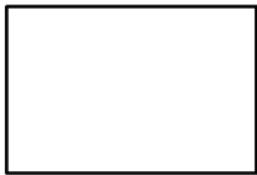
#### Out

```
0
4
6
4
```

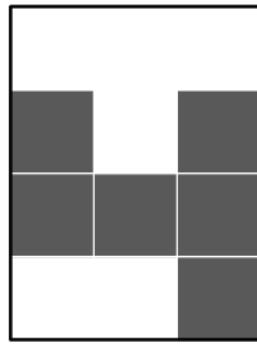
### 样例解释 #1

如图，四组数据最优策略的情况。

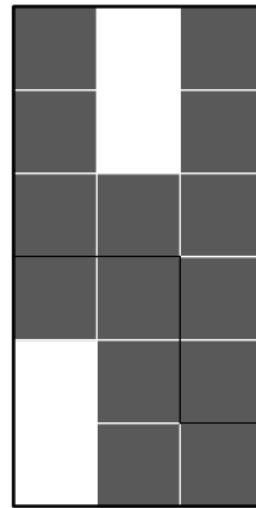
$m = 2$



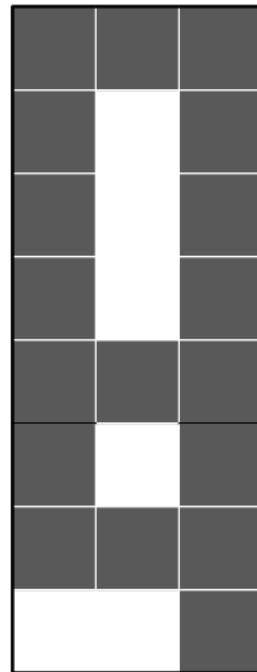
$m = 4$



$m = 6$



$m = 8$



## 【数据范围】

对于所有数据有  $1 \leq T \leq 100$ 。

分值	数据范围
20	$1 \leq m \leq 20$
20	$1 \leq m \leq 10^3$
20	$1 \leq m \leq 10^5$
20	$1 \leq m \leq 10^9$
20	$1 \leq m \leq 10^{18}$

## [K] 抽卡打怪

### 【题目描述】

你面前有  $n$  堆不同的卡牌，其中每堆均有  $m$  张牌。每一张牌都有自己的能力值  $a_{ij}$ ，其中  $a_{ij}$  是第  $i$  堆牌的第  $j$  张。

你可以进行  $k$  次抽卡操作，对于每一次操作：

- 你可以抽取一牌堆顶部的两张牌，选取一张加入自己的手牌中，并将剩下的牌放置至该牌堆顶部。

在  $k$  次抽卡操作之后，你将会计算手牌中  $k$  张牌的总能力值（即所有牌的能力值之和），你想让总能力值尽可能大。

请问当手牌总能力值最大时，总能力值将会是多少？

### 【输入格式】

一行 3 个整数  $n, m, k$ ，分别代表牌堆的数量、每一个牌堆的牌数和能进行抽卡的次数。

接下来  $n$  行，每行  $m$  个整数，其中  $a_{ij}$  表示第  $i$  堆卡牌从牌堆顶部往下数的第  $j$  张。

### 【输出格式】

一行一个整数，表示最大的总能力值。

### 【样例】

#### 样例 1

In

```
5 6 10
1 1 4 5 1 4
1 2 3 4 5 6
6 5 4 3 2 1
4 5 1 4 1 1
1 4 2 8 5 7
```

Out

50

## 样例说明 # 1

$k=10$

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

$\text{total}=50$

## 【数据范围】

- $1 \leq n, m \leq 200$
- $1 \leq k \leq \min(n \times m, 1000)$
- $1 \leq a_{i,j} \leq 10^4$ .

分值	数据范围
30	$1 \leq n, m \leq 20$
70	无额外限制

## [L] 「星光渐明之时」

### 【题目描述】

你所拥有的，是一片由  $n$  颗星星组成的星空，每颗星星都有一定的亮度（下标为  $i$  的星星有  $a_i$  的亮度）。现在，你需要回答  $q$  次询问，每个询问属于以下两种情况之一：

1. 给定星星的下标  $i$ ，将第  $i$  颗星星的亮度提升至原来的两倍。
2. 给定星空区域  $[l, r]$ ，计算最少需要多少次操作才能使该区域的星星亮度形成从左到右单调不减的序列。该询问并不会改变星星的亮度。

### 【输入格式】

第一行包含两个整数  $n$  和  $q$ ，分别表示星星数量和查询次数。

第二行包含  $n$  个整数  $a_1, a_2, \dots, a_n$ ，表示初始时每颗星星的亮度。

接下来  $q$  行，每行描述一个查询，格式如下：

- 若查询类型为 1，则该行格式为 1  $i$ ，表示将第  $i$  颗星星的亮度翻倍。
- 若查询类型为 2，则该行格式为 2  $l$   $r$ ，表示询问在区间  $[l, r]$  内，最少需要多少次亮度翻倍操作，才能使该区间内亮度单调不降。

### 【输出格式】

对于每个类型为 2 的查询，输出一个整数，表示使对应区间亮度单调不降所需的最少操作次数。

每个答案占一行，按查询顺序输出。

### 【样例】

#### 样例 1

##### In

```
10 3
5 2 7 3 2 9 6 3 3 5
2 1 10
1 1
2 1 10
```

##### Out

```
27
36
```

## 样例 2

### In

```
8 6
6712 8762 198 999 410 10001 3614 507
2 1 8
1 1
2 3 5
1 2
2 2 6
2 1 8
```

### Out

```
29
2
22
35
```

## 样例说明 #1

初始亮度	5	2	7	3	2	9	6	3	3	5	
翻倍次数	0	2	1	3	4	2	3	4	4	4	TOTAL = 27
最终亮度	5	8	14	24	32	36	48	48	48	80	

初始亮度	10	2	7	3	2	9	6	3	3	5	
翻倍次数	0	3	2	4	5	3	4	5	5	5	TOTAL = 36
最终亮度	10	16	28	48	64	72	96	96	96	160	

## 【数据范围】

- $1 \leq n \leq 10^5$
- $1 \leq q \leq 10^5$
- $1 \leq a_i \leq 10^9 \quad (1 \leq i \leq n)$
- 对所有查询满足  $1 \leq l \leq r \leq n$ .

分值	数据范围
8	$1 \leq n, m \leq 10^4$
24	只包含查询类型 2
68	无额外限制