# A 月相

时空限制：1s 256MB

### 文件名

moon.in/moon.out/moon.cpp

## 题意

果果最近在观察月相, 他把月亮的大小分为 $0\sim 15$ 这 $16$ 个大小等级, 月亮会在每个月先每天放大 $1$, 从 $0$ 缓慢变大到 $15$, 然后再每天减小 $1$, 从 $15$ 逐渐变成 $0$ , 以此循环往复。 果果连续观察了 $2$ 天月亮的大小, 果果想知道下一天月亮的大小是多少。

## 输入格式

第一行输入两个整数 $a, b$, 表示果果观察的月亮的大小, 分别是第一天、第二天的结果。

## 输出格式

一行一个整数，表示下一天的月亮的大小。

## 输入样例

```
15 14
```

## 输出样例

```
13
```

## 样例解释

前两天分别是 $15, 14$, 故果果知道月亮在逐渐变小，所以下一天是 $13$

## 数据范围

对于 $60\%$ 的数据有 $1 \le a, b \le 14$ 。

对于 $100\%$ 的数据有 $0 \le a, b \le 15$ 。

# B 间隔

时空限制：2s 256MB

## 文件名

interval.in/interval.out/interval.cpp

# 题意

果果现在有一个从小到大排好序的整数序列 $A$。

他希望插入一些整数, 使得相邻两两之间的差值一样大。

果果想知道最少插入多少个整数才能满足要求, 如果不能, 则输出 $-1$ 。

# 输入格式

第一行一个整数 $T$, 表示数据组数。

每组数据第一行输入一个整数 $n$, 表示序列 $A$ 的大小。

每组数据第二行输入 $n$ 个整数, 表示 $A$ 数组中的数字 $A_i$ 。

# 输出格式

对于每组数据输出一行一个整数，即插入的最小数字个数。

# 输入样例

```
3
4
3 7 11 19
3
3 8 10
5
3 9 15 21 21
```

# 输出样例

```
1
5
-1
```

# 样例解释

```
对于样例1，可以填入一个15，使得(3,7,11,15,19)相邻两两差值为4。
对于样例2，可以填入(4,5,6,7,9)，使得(3,4,5,6,7,8,9,10)相邻两两差值为1。
对于样例3，无论怎么插入都无法使数组相邻两两间差值相同。
```

## 数据范围

对于 $30\%$ 的数据有 $1 \le n \le 3$ 。

对于 $60\%$ 的数据有 $1 \le n \le 1000$ 。

对于 $100\%$ 的数据有 $1 \le n \le 10^5, 0 \le A_i \le 10^6, 1 \le T \le 10$ 。

# C 密码锁

时空限制：2s 256MB

### 文件名

lock.in/lock.out/lock.cpp

## 题意

果果家的密码锁, 输入密码的位置是 $N$ 行 $M$ 列的, 每个位置都是一个不同字母、数字或符号的按钮。

果果发现, 每次点击一个按钮，都会发出声音。可以通过辨识前后两个声音的间隔, 来判断前后两个按钮的最大距离。 果果把听到的声音间隔分析了一下, 计算出了每个位置离前一个按钮的最大距离。距离是指曼哈顿距离, 即上下左右相邻的按钮距离为 $1$。当然，有时候输入密码的人会停顿非常久，所以让果果计算出来的距离非常大, 甚至比最远的两个按钮还大, 但这只是可能的最大距离, 计算时应该予以考虑。

已知密码长度为 $t$, 果果不知道密码输入的起始位置, 但果果想知道通过这样的计算, 有多少种可能的密码。结果对 $10^9+7$ 取模。

## 输入格式

第一行输入三个整数 $N, M, t$, 表示密码锁的按钮行列数，和密码长度。

第二行输入 $t-1$ 个整数, 第 $i$ 个整数 $a_i$ 表示按下的第 $i+1$ 个位置和第 $i$ 个位置的距离。

## 输出格式

一行输出一个整数，即密码可能的数量。

## 输入样例

```
1 3 2
1
```

## 输出样例

```
7
```

## 样例解释

```
开始位置可能是 (1,1),(1,2),(1,3)。
如果第一个按钮和第二个按钮距离为 0 ，则密码为 (1,1)->(1,1),(1,2)->(1,2),(1,3)->(1,3)，有三种不同的密码。
如果第一个按钮和第二个按钮距离为 1 ，则密码为 (1,1)->(1,2),(1,2)->(1,1),(1,2)->(1,3),(1,3)->(1,2)，有四种不同的密码。
共计有 7 种不同的密码。
```

## 数据范围

对于 $20\%$ 的数据有 $n=1$ 或 $m=1$ 。

另有 $20\%$ 的数据有 $1\le n,m\le 3$ 。

对于 $60\%$ 的数据有 $1\le n,m\le 10$ 。

对于 $100\%$ 的数据有 $1\le n,m\le 25, 2\le t\le 1000, 0\le a_i\le 10^9$ 。

# D 上街

时空限制：12s 256MB

### 文件名

street.in/street.out/street.cpp

## 题意

果果注意到家周围的红绿灯排布可以看成一个 $n\times m$ 的方格图, 位置从左上角的 $(1,1)$ 到右下角的 $(n,m)$, 右上角为 $(1,m)$ 。地图是上北下南左西右东的。

果果计算好了每个路口向下走到下一个路口的时间 $d_i$ 和向右走到下一个路口需要花费的时间 $e_i$ 。

在每个路口, 向前或向左走只有在绿灯时才可以前进，向右走则可以忽略红绿灯而直接向前走, 通过路口的时间可以忽略不计。某些路口没有加装红绿灯, 这些路口可以选择任意方向前进而不需要等待。

每个路口的红绿灯有自己独立的时间, 分别是东西（左右）方向的 $a_i$ 秒绿灯和南北（上下）方向的 $b_i$ 秒绿灯，东西方向为绿灯时南北方向为红灯, 南 北方向为绿灯时同理, 初始时所有红绿灯都在南北方向红灯第0秒（如第 $\mathrm{i}$ 个路口, 初始之后是南北方向是 $a_i$ 秒红灯，随后是 $b_i$ 秒绿灯, 然后是 $a_i$ 秒红 灯，以此类推）。为了简化问题, 设定 $t=a_i+b_i$, 对于所有红绿灯, 红灯加绿灯的时间是一样的。如果 $a_i=b_i=0$, 则表示该路口未加装红绿灯。

果果喜欢欣赏沿途的风景, 唯独不喜欢等红绿灯, 所以对他来说, 等待红绿灯的时间在他看来时间会流逝缓慢许多。而且, 他不会在一条道路的中途 停下等时间, 除非是等红绿灯。

他认为的花费代价 $=$ 红绿灯等待时间 $\times 10+$ 路途花费时间（不含等红绿灯)

果果一开始在位置 $(1,1)$ 点朝下（南）骑行（等待红绿灯），他希望到达路口 $(x_e,y_e)$ (任意方向均可）, 想知道最小花费的代价是多少。

## 输入格式

第一行输入三个整数 $n, m, t$, 表示地图的大小, 和红绿灯的总时长 $(a_i + b_i)$ 。

第二行输入两个整数 $x_e, y_e$, 表示果果要到达的终点。

随后 $n * m$ 行，每行输入四个整数 $a_i, b_i, d_i, e_i$, 分别为位置 $(i/m + 1$ (整除), $i\%m + 1$ ) 的红灯时间、绿灯时间, 向下到下一个路口的距离和向右到下一 个路口的距离。对于边缘位置给出的，通向地图之外的 $d_i, e_i$ 应该忽略。

## 输出格式

一行输出一个整数，即花费的最小代价。

## 输入样例

```
2 3 30
2 3
15 15 15 30
15 15 60 15
0 0 100 0
15 15 0 70
15 15 0 30
20 10 0 0
```

## 输出样例

```
270
```

## 样例解释

```
有几条可选的路线，我们分别来分析一下：
一开始需要在(1,1)等15秒的红灯，代价是150。
路线1：(1,1) -> (1,2) 路程花费时间30， (1,2)->(1,3) ，正值东西方向红灯，等待15秒，代价150，路程花费15，(1,3)->(2,3)，此时因为是右转，不需要等待红绿灯（而且这里也未加装红绿灯），路程花费100，共计代价150+30+150+15+100 = 445 。
路线2：(1,1) -> (1,2) 路程花费时间30， (1,2)->(2,2) 右转忽略红绿灯，路程花费60，(2,2)->(2,3)，此时时间为105，正值南北方向绿灯，路程花费30，共计代价 150+30+60+30 = 270 。
路线3：(1,1) -> (2,1) 路程花费时间15，(2,1)->(2,2)，正值南北方向红灯，等待15秒，代价150，路程花费70，(2,2)->(2,3) 正值东西方向红灯，等待5秒，代价50，路程花费30，共计代价 150+15+150+70+50+30=465 。
可能还有一些其他路线，如(1,1)->(2,1)->(2,2)->(1,2)->(1,3)->(2,3)，代价更高一些，综上，最小代价为270 。
```

## 数据范围

对于 $30\%$ 的数据, $n, m \le 3$ 。

另有 $10\%$ 的数据, $n \le 1$ 或 $m \le 1$

对于总量 $30\%$ 的数据, $a_i, b_i = 0$ 。

以上三种数据共占 $60\%$ 。

对于 $100\%$ 的数据有 $1 \le n, m \le 200, 0 \le a_i, b_i \le t \le 60, 0 \le d_i, e_i \le 10^4$ 。

数据规模阶梯型增大。