

欢迎参加北京化工大学 2018 年 10 月程序设计竞赛！

月赛为基础算法题，适合 ACM 入门队员做，现场赛铜奖及以上队员不宜参加
竞赛网址：<http://39.105.0.22/contest.php>

自行注册即可参赛，赛后在原 oj 安排补题，并发布解题报告

问题 A: 水题的 ZZH

ZZH 是北化 acm 界新生代的大佬，每天会和无数来自全球各地的大佬进行交流。但是，ZZH 每天水群的时间有限。ZZH 想尽可能多的和更强的大佬进行交流。他把每个大佬的实力用一个数字 Ω 表示出来，并且每天和 Ω 总和最高的几位大佬进行交流。但是 ZZH 正在忙着水群，于是把这个问题交给了你。事成之后，ZZH 会给你 $2147483647\%1$ 的金币作为奖励

输入

单组数据

第一行两个整数 n ， m ，分别代表今天 ZZH 水群的时间（以分计算）和今天想要与 ZZH 交流的大佬的数量（ ≤ 2000 ）。

之后 m 行，分别有两个数字，代表和第 i 个大佬交流花费的时间（ $< n$ ），以及大佬的 Ω （ < 1000 ）。

输出

今天与 ZZH 交流的大佬的 Ω 的值的总和。

样例输入

```
70 3
71 100
69 3
1 2
```

样例输出

```
5
```

问题 B: 积木

我懒得编故事了 m 哥牛逼！

有 n 个积木，第 i 个积木的高度为 $a[i]$ ，求每个积木前面有几个比它矮的积木

输入

多组输入，每组由两行组成，第一行输入 n ，第二行输入 n 个数，为 n 个积木的高度 $a[i]$

//(1<=n<=100000)(1<=a[i]<=1000000000)

输出

每组数据输出 n 个数，第 i 个数为第 i 个积木前面比他矮的积木的个数

样例输入

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

样例输出

```
0 1 2 3 4
```

问题 C: 云计算

北化 ACM 社团有 n 台云服务器，每台服务器都有一个使用期限，第 i 台服务器还能使用 R_i 天。现在有 m 个任务需要部署到云服务器上运行，第 j 个任务需要运行 T_j 天。第 j 个任务能够部署在第 i 台服务器上时当且仅当 $T_j \leq R_i$ ，并且每台服务器在其使用期限内总共只能运行一个任务。即使 $T_k + T_j \leq R_i$ ，你也无法将第 k 个和第 j 个任务一起部署到第 i 台服务器上，否则这题将失去其签到题的作用。为了能充分利用云服务器，现在需要计算最多能部署多少任务。而你作为北化 ACM 的希望,这个问题需要由你来解决。

输入

单组数据

第一行两个整数代表 n 和 m

第二行 n 个整数代表 R_1, R_2, \dots, R_n

第三行 m 个整数代表 T_1, T_2, \dots, T_m

$1 \leq n, m, T_j, R_i \leq 100000$

输出

输出一个整数代表最多有多少个任务能够部署到云服务器上

样例输入

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

4 6 6 2 5

样例输出

3

问题 D: 罗 dalao 的密码

众所周知，罗 dalao 十分的厉害。有一天，罗 dalao 发明了一种密码，用来给文档加密(可认为文档是一个长度为 n 的字符串 s)。

罗 dalao 要对 $s[l]...s[r]$ 加密，首先要反转区间 $s[l]...s[r]$ (闭区间)，然后使 $k=(r-l+1)/2$ (向下取整)，再反转区间 $s[l]...s[l+k-1]$ 和 $s[l+k]...s[r]$ ，以此类推，一直到区间长度都为 1 为止。(加密整个文档相当于加密区间 $s[1]...s[n]$)

有一天你找到了一篇罗 dalao 加密过的文档，现在要你求出某个位置的字符加密前的位置。

输入

第一行输入整数 $T(0 < T \leq 1000)$ ，表示数据组数，接下来 T 行，每行两个整数 n 和 $p(1 \leq p \leq n \leq 1e18)$ ，分别表示文档长度和询问的位置输出

输出 T 行，每行一个整数，表示每组数据中加密后 p 位置的字符的原位置

样例输入

4

4 1

4 2

4 3

4 4

样例输出

3

4

1

2

问题 E: 罗 dalao 的小电影

罗 dalao 非常的厉害，于是会有很多人向他请教一些问题，而他每天有 L 分钟专门用来解答问题。由于来请教的人很多，所以要找罗 dalao 请教问题，需要预定。

于是，罗 dalao 知道每天会有 n 个人来请教问题，第 i 个到来的人在第 a_i 分钟到来，罗 dalao 解答问题要用 t_i 分钟(罗 dalao 精明的安排保证 $a_i + t_i \leq a_{i+1}$ 和 $a_n + t_n \leq L$)，然而罗 dalao 想找空闲的时间看小电影，罗 dalao 看每一部小电影都要从头看到尾，期间不能有中断，每部小电影的时长为 m ，罗 dalao 想知道他每天最多可以看多少部小电影，现在请你帮他计算一下。

输入

单组

第一行 3 个整数： n ， L 和

m ($0 \leq n \leq 1e5, 1 \leq L \leq 1e9, 1 \leq m \leq L$) 接下来 n 行，每行

两个整数 a_i 和 t_i ($0 \leq a_i \leq L-1, 1 \leq t_i \leq L$) 输出

输出一个整数，表示罗 dalao 最多能观看多少部小电影

样例输入

```
2 11 3
```

```
0 1
```

```
1 1
```

样例输出

```
3
```

问题 F: 数列

故事不编了，m 哥牛逼！

给出一个长度为 n 的数列 a ，现在有如下三种操作：

- 1、 1 p x: 将 $a[p]$ 更改为 x
- 2、 2 l r: 求 $a[l]$ 到 $a[r]$ 的最大值与最小值的差
- 3、 3 l r: 求 $a[l]$ 到 $a[r]$ 的和现在给你一些操作，请你求出相应的数值。

输入

第一行输入一个数字 T 表示数据组数 ($1 \leq T \leq 10$)

接下来 T 组数据，每组数据第一行为两个整数 n 、 m ($1 \leq n, m \leq 1e5$)，分别代表 a 的长度和操作数，下一行 n 个整数表示 $a[1]$ 到 $a[n]$ ($0 \leq a[i] \leq 1000$)，接下来 m 行，每行一个如题目给出的输入格式的操作
($1 \leq p, l, r \leq n, 0 \leq x \leq 1000$)

输出

对于每组数据对每一个操作 2 或 3，输出一个整数表示结果

样例输入

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

样例输出

```
1
6
15
5
7
1
```

问题 G: Leo 的简单规律题

Leo 喜欢吃烧烤，而且只吃三种，羊肉串，青菜，火腿肠，她吃烧烤之前喜欢把烧烤摆成一长串，而且还有特别的摆放规则，摆错了她这一顿就不吃了。规则有：1.对于连续的三个烤串，不能三个烤串都是同一种。

2.对于连续的三个烤串，如果三种烤串都有，青菜不能放在中间（在吃肉中途吃菜是会令人自闭的）

3.依旧是对于连续的三个烤串，第一串和第三串都是青菜的话，中间不能是羊肉或者火腿肠

（她的意思是吃素减肥）

现在有三种烤串都有无数根，她想摆出长度为 n 的烤串，请你求出方案数（由于结果可能会很大，所以只需给出答案模 1000000007）

输入

第一行有一个数字 T ($T < 100$)，表示有 T 组

数据接下来的 T 行，每行一个数字 n

($1 \leq n \leq 1010$)

输出

对于每一组数据，输出摆出长度为 n 的烤串的方案数

样例输入

```
4
2
3
4
6
```

样例输出

```
9
20
46
244
```

问题 H: Leo 的简单数列

Leo 有两个数列，它们的长度都是 n 在最开始， a 数列里都是 0， b 数列是 Leo 随机填充的 n 个数（数字为 $1-n$ ，保证 b 数列是

$1-n$ 的一个排列）现在

Leo 可以进行两种操作

1. add $L\ R$ 表示她给 a 数列从 $a[L], a[L+1], \dots, a[R]$, 都加一

2. query $L\ R$ 表示她想知道当 $L \leq i \leq R$ 时 $a[i]/b[i]$ （向下取整）的所有项的总和当然 Leo 在数学这一块还不是很熟练，上次月考数学才 110 分，所以想请你帮她完成这些操作

输入

多组数据

对于每组数据，第一行有 2 个数字 n, q ，表示 a 与 b 的长度和操作数。 $1 \leq n, q \leq 100000$ 第二行有 n 个数字，表示数列 b 的初始状态接下来的 q 行，每行一个操作，具体操作格式如题目所给 $1 \leq L \leq R \leq n$

输出

对于每一个 query 操作，输出一行表示求和的结果。

样例输入

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

样例输出

```
0
0
2
2
4
1
```