

## Problem A. Sumo and Keyboard-Cat

Time limit: 1 second  
Memory limit: 512 megabytes

*Sumo* 家的猫非常喜欢滚键盘，每次 *Sumo* 开着电脑离开一小会儿，回来的时候都能在屏幕上看到一串神秘代码。

今天，*Sumo* 又得到了这么一串神秘代码。当他盯着屏幕上这串东西沉思的时候，突然发现它只包含大小写英文字母，又想到自己的键盘这几天两个 *Shift* 键都坏掉了——也就是说只能通过 *CAPSLOCK* 键来切换英文大小写了。于是他很好奇，这次滚键盘的过程中，他家的猫至少摁了几次 *CAPSLOCK* 键呢？

*Sumo* 记得他刚离开的时候，键盘的大写锁定是开的（也就是说输的是大写字母）。

### Input

仅包含一个字符串，表示 *Sumo* 家的猫摁出的神秘代码。

输入保证字符串仅包含大小写英文字母，且长度不超过  $10^5$ 。字符串的末尾有 '\n' 换行。

### Output

仅一个整数，表示这次滚键盘的过程中，*Sumo* 家的猫摁 *CAPSLOCK* 键的最少次数。

### Examples

Sample Input	Sample Output
ZUCCAcmlab	3
MEOWMEOWMEOWGiveMeCatFoodMEOWMEOWMEOW	8

## Problem B. Sumo and His Followers

Time limit: 2 seconds  
Memory limit: 512 megabytes

*Sumo* GG is very popular in the laboratory. Many people come to ask *Sumo* GG for questions.

Now there are  $n$  people in line to ask him for advice, as the time for  $i$ -th people to ask him questions is  $t_i$ . In order not to affect everyone's lunch, please line up for  $n$  people so that the average waiting time of  $n$  people is the minimum.

### Input

The first line contains  $T$  ( $1 \leq T \leq 10$ ) —the number of test cases.

For each test case, the first line contains a single integer  $n$  ( $1 \leq n \leq 10^5$ ), the number of people waiting for asking a question.

The next line contains  $n$  integers  $t_1, t_2, \dots, t_n$  ( $1 \leq t_i \leq 1000$ ), the time of  $i_{th}$  people required to ask a question.

### Output

For each of the test cases, output a single integer, the minimum average waiting time of  $n$  people (accurate to two decimal places).

### Examples

Sample Input	Sample Output
2	0.00
1	291.90
5	
10	
56 12 1 99 1000 234 33 55 99 812	

## Problem C. Sumo and Virus

Time limit: 1 seconds  
Memory limit: 512 megabytes

*Sumo* 生活的小镇有  $m$  个居民, 小镇的生活和谐而美好。但是, 有一天, 可怕的事情发生了.....

这个故事要从一只蝙蝠讲起。

一个月黑风高的夜晚, *Sumo* 家中闯入了一只迷路的蝙蝠, *Sumo* 在驱赶蝙蝠的过程中一不小心被蝙蝠咬伤了。结果就悲剧了, 他感染了一种传染性很强的病毒。现在发现这种病毒的传染情况如下:

1. 一个病患每天可以传染  $x$  个未被感染的人;
2. 潜伏期为 7 天, 期间不发病也不传染别人;
3. 第 8 天开始发病, 并且可以传染;
4. 第 14 天, 被治愈 (不再具有传染能力);
5. 治愈之后具有抵抗力, 不会被传染。

问: 从 *Sumo* 感染病毒开始算第一天, 第  $n$  天小镇上有几个传染者 (指具有传染能力的人)?

### Input

第一行包含一个正整数  $T(1 \leq T \leq 100)$ , 表示一共有  $T$  组数据。

第二行包含三个正整数  $x(1 \leq x \leq 10^5)$ ,  $m(1 \leq m \leq 10^5)$ ,  $n(1 \leq n \leq 10^5)$ , 含义如题目所描述。

### Output

输出一个整数代表第  $n$  天有几个传染者 (指具有传染能力的人)。

### Examples

Sample Input	Sample Output
3	0
3 4 7	1
3 4 12	3
3 4 16	

## Problem D. Sumo and Easy Sum

Time limit: 1 second  
Memory limit: 512 megabytes

Give a positive integer  $K$  and the following conditions:

$$\begin{cases} a_1 = a_2 = 1 \\ a_i = a_{i-1} + a_{i-2} \quad (3 \leq i \leq n) \\ \lim_{n \rightarrow \infty} \frac{a_{n-1}}{a_n} = \frac{\sqrt{5}-1}{2} \\ S_n = \frac{1}{K} + \frac{1}{K^2} + \frac{2}{K^3} + \frac{3}{K^4} + \frac{5}{K^5} + \frac{8}{K^6} + \cdots + \frac{a_{n-1}}{K^{n-1}} + \frac{a_n}{K^n} \end{cases}$$

Please help *Sumo* write a program to calculate  $\lim_{n \rightarrow \infty} S_n$ , the result should be expressed as a simple fraction.

### Input

The first line gives an integer  $T (1 \leq T \leq 10^4)$ , indicates that the following  $T$  test cases will be given.

Next  $T$  lines, each line gives a positive integer  $K (2 \leq K \leq 10^4)$ .

### Output

For each test case, print a line to express the answer.

### Examples

Sample Input	Sample Output
2	2
2	3/5
3	

### Note

6/5, 2, 3/5 are simple fractions, but 2/1, 10/5, 1/1 not.

## Problem E. Sumo and Group Activity

Time limit: 1 second  
Memory limit: 512 megabytes

Sumo 是一个社交达人，他经常辗转各地开展 party。

将社区想象为一个坐标轴，给定许多个可以开展 party 的坐标点，以及 Sumo 的家的坐标点。Sumo 接下来要开展  $m$  次 party，因此还需要选择  $m$  个地点。

Sumo 希望知道有多少种选取地点的方案，使得这些方案满足以下条件：

- 若第  $i$  个选择地点的坐标为  $p_i$ ，那么  $p_i \neq p_{i-1} (2 \leq i \leq m)$ 。
- 若  $b$  为家的坐标点，那么  $|p_i - p_{i+1}| < |p_i - b| (1 \leq i \leq m - 1)$ ，即当前地点到下一个地点的距离要严格小于当前地点到家的距离。
- 可以选择自己的家作为 party 地点。

请告诉 Sumo 有多少种选取 party 地点的方案，答案可能过大，请输出答案对  $10^9 + 7$  取模后的结果。

两种方案  $(k_1, k_2, k_3 \dots k_n)$  和  $(p_1, p_2, p_3 \dots p_n)$ ，若存在  $(1 \leq i \leq n)$  且  $(k_i \neq p_i)$ ，则认为两个方案是不同的。

### Input

输入的第一行包含四个空格分隔的整数  $n, a, b, m (2 \leq n \leq 5000, 1 \leq m \leq 5000, 1 \leq a, b \leq n, a \neq b)$ 。 $n$  表示有  $n$  个备选地点， $a$  表示 Sumo 最近一次开 party 的地点序号， $b$  表示 Sumo 家的地点序号， $m$  表示还要举办 party 的次数。

接下来一行有  $n$  个数  $x_1, x_2, x_3 \dots x_n (1 \leq x_i \leq 10^9)$ ，其中  $x_i$  代表序号为  $i$  的地点的坐标，保证坐标两两不同。(两个地点之间的距离即为坐标之差)

### Output

由于答案可能过大，因此只需要输出一个整数，表示总数对  $10^9 + 7$  取模后的结果。

### Examples

Sample Input	Sample Output
5 4 3 1 25 18 20 13 7	2
5 2 4 2 7 13 18 20 25	2
5 3 4 1 7 13 18 20 25	0

### Note

对于样例 1，Sumo 可以选 5 号地点或 2 号地点。

对于样例 2，Sumo 可以选 (1 号地点, 2 号地点) 或者 (1 号地点, 3 号地点)。若 Sumo 一开始选 3 号地点那他接下来就没地方可以选了。

## Problem F. Sumo and Luxury Car

Time limit: 1 second  
Memory limit: 512 megabytes

As we all know, *Sumo* has a lot of luxury cars, Maserati, Porsche, Lincoln and so on. Countless cars are parked in his garage. He is worried about what kind of car he drives every day. Can you help him?

*Sumo* has a total of  $n$  cars. Every day, he will choose any number of cars from the  $n$  cars (the number of cars cannot be 0) to form a team, and then choose one from this team to drive himself. How many options are there?

**If the selected team set is different or the car he chooses is different, it is considered to be two different plans.**

### Input

The first line of the input is a single integer  $T$  ( $1 \leq T \leq 10$ ) which is the number of test cases.  $T$  test cases follow.

Each test case has a single integer  $n$  ( $1 \leq n \leq 10^9$ ), representing the total number of luxury cars in *Sumo*.

### Output

For each test case, print a single line containing an integer modulo  $10^9 + 7$ .

### Examples

Sample Input	Sample Output
2	1
1	4
2	

### Explanation

In the second sample, *Sumo* has four situations to drive:

1. Choose cars with numbers 1 and 2, drive the car with number 1,
2. Choose cars with numbers 1 and 2, drive the car with number 2,
3. Choose the car with numbers 1, drive the car with number 1,
4. Choose the car with numbers 2, drive the car with number 2.

So the answer is 4.

## Problem G. Sumo and Robot Game

Time limit: 1 second  
Memory limit: 512 megabytes

*Sumo* 正在参加一个机器人比赛，这个比赛的规则是这样的：在一个一维坐标轴上操控一个初始位置为 0 的机器人，机器人每次只能向左或者向右移动一个单位距离。每位参赛者都需要上传一个长度恰好为  $x$  的仅由 'L' 和 'R' 组成的执行序列，'L' 表示向左移动，'R' 表示向右移动（例如 "LRLLR" 即为长度为 5 的执行序列，机器人的运动轨迹为  $-1, 0, -1, -2, -1$ ），随后测试系统会严格按照执行序列给出的命令顺序模拟机器人的行动。

在坐标轴上分布着  $n$  个得分点（由正数给出）或扣分点（由负数给出），当机器人**第一次**到达某个得分点或扣分点时，它会获得或扣去相应的分数（初始分数为 0），在每个人的执行序列执行完毕后系统会给出最终得分。

*Sumo* 想让你帮助他设计一个执行序列来获得尽可能多的分数，由于序列可能过长，因此你只需要告诉 *Sumo* 最多能拿到多少分数。

### Input

第一行包含两个整数  $n(1 \leq n \leq 10^5)$ ，表示一共有  $n$  个得分点或扣分点， $x(1 \leq x \leq 10^9)$  表示执行序列的长度。

第二行给出  $n$  个整数  $p_i(p_i \neq 0)$  表示第  $i$  个点在轴上的位置，且  $-10^9 \leq p_1 < p_2 < \dots < p_{n-1} < p_n \leq 10^9$ 。

第三行给出  $n$  个整数  $v_i(-10^4 \leq v_i \leq 10^4)$  正数表示得分点，负数表示扣分点，其绝对值表示当第一次到达该点时所应获得或扣去的分数值。

### Output

输出一个整数代表 *Sumo* 有可能得到的最大分数。

### Examples

Sample Input	Sample Output
6 6 -3 -2 -1 1 2 3 1 -1 4 5 1 4	14

### Explanation

样例中机器人能获得最高分数的执行序列可能为 "LRRRRL"，其运动轨迹为  $-1, 0, 1, 2, 3, 2$ ，因此可以获得坐标为  $-1, 1, 2, 3$  的点上的分数，共  $4 + 5 + 1 + 4 = 14$  分。

## Problem H. Sumo and Electricity(Easy)

Time limit: 1 second  
Memory limit: 512 megabytes

注意: *Easy* 与 *Hard* 版本的不同之处在于  $w_i$  的条件限制, *Easy* 中有且仅有一个  $w_i$  满足  $w_i = -1$ 。

*Sumo* 有  $n$  个核电站, 每个核电站都有自己的耗电量  $w_i (1 \leq i \leq n)$ 。这些核电站之间通过  $m$  条电缆相连, 电缆的传输功耗为两个核电站耗电量之间的异或  $\oplus$  (XOR) 值。

但是核电站功率十分不稳定, 因此 *Sumo* 并不知道部分核电站目前的功耗是怎样的, 但是它可以选择调整这些核电站功耗的大小。

因为电缆的功耗远大于核电站的功耗, 因此 *Sumo* 希望可以优先保证在所有电缆功耗总和尽可能低的情况下, 尽量降低所有核电站的功耗总和, 希望你能帮助 *Sumo* 为功耗未知的核电站设置功耗, 从而满足以上的条件。

### Input

第一行输入整数  $n, m (1 \leq n \leq 500, 1 \leq m \leq 2000)$  代表有  $n$  个核电站,  $m$  条电缆。

接下来的一行中给出  $n$  个整数  $w_i (-1 \leq w_i < 2^{31})$  代表第  $i$  个核电站的功耗,  $w_i = -1$  代表这个核电站的功耗未知, **本题中有且仅有一个  $w_i$  满足  $w_i = -1$ 。**

接下来  $m$  行, 每行输入两个数  $u_i, v_i (1 \leq u_i < v_i \leq n)$  表示该条电缆所连接的两个核电站的编号, 数据保证两个核电站之间不会有多条电缆连接。

### Output

第一行输出一个整数表示电缆功耗总和。

第二行输出一个整数表示在电缆功耗总和尽可能低的情况下, 核电站的功耗总和。

### Examples

Sample Input	Sample Output
3 2 2 -1 0 1 2 2 3	2 2



## Problem I. Sumo and Electricity(Hard)

Time limit: 1 second  
Memory limit: 512 megabytes

注意: *Easy* 与 *Hard* 版本的不同之处在于  $w_i$  的条件限制, *Hard* 中存在多个  $w_i$  满足  $w_i = -1$ 。

*Sumo* 有  $n$  个核电站, 每个核电站都有自己的耗电量  $w_i (1 \leq i \leq n)$ 。这些核电站之间通过  $m$  条电缆相连, 电缆的传输功耗为两个核电站耗电量之间的异或  $\oplus$  (XOR) 值。

但是核电站功率十分不稳定, 因此 *Sumo* 并不知道部分核电站目前的功耗是怎样的, 但是它可以调整这些核电站功耗的大小。

因为电缆的功耗远大于核电站的功耗, 因此 *Sumo* 希望可以优先保证在所有电缆功耗总和尽可能低的情况下, 尽量降低所有核电站的功耗总和, 希望你能帮助 *Sumo* 为功耗未知的核电站设置功耗, 从而满足以上的条件。

### Input

第一行输入整数  $n, m (1 \leq n \leq 500, 1 \leq m \leq 2000)$  代表有  $n$  个核电站,  $m$  条电缆。

接下来的一行中给出  $n$  个整数  $w_i (-1 \leq w_i < 2^{31})$  代表第  $i$  个核电站的功耗,  $w_i = -1$  代表这个核电站的功耗未知。

接下来  $m$  行, 每行输入两个数  $u_i, v_i (1 \leq u_i < v_i \leq n)$  表示该条电缆所连接的两个核电站的编号, 数据保证两个核电站之间不会有多条电缆连接。

### Output

第一行输出一个整数表示电缆功耗总和。

第二行输出一个整数表示在电缆功耗总和尽可能低的情况下, 核电站的功耗总和。

### Examples

Sample Input	Sample Output
4 4 1 -1 -1 10 1 3 1 2 2 3 2 4	11 13

### Explanation

样例中当 *Sumo* 将 2, 3 两个核电站的功耗设置为 1, 1 时, 电缆总功耗为:  $(1 \oplus 1) + (1 \oplus 1) + (1 \oplus 1) + (1 \oplus 10) = 11$ , 总功耗最小, 核电站的功耗总和为 13。

## Problem J. Sumo and Balloon

Time limit: 1 second  
Memory limit: 512 megabytes

*Sumo* 喜欢吹气球，更喜欢吹炸气球。

今天，他准备了一面粘满密密麻麻钉子的墙（尖锐面朝着 *Sumo*），站在离墙一定距离的地方朝着垂直于墙面的方向吹气球。他想知道，从他往气球里吹起开始计时，气球会在什么时间点爆炸。

为了方便计算，我们对问题进行简化：

- 气球被视为是一个理想的球体，可以无限增大。
- 气球的吹气口位于气球的边缘，吹气时，气球的增大会导致球心顺着吹气的方向平移，不考虑重力对气球下垂的影响。
- 在整个吹气过程中，*Sumo* 的嘴一直固定不动，匀速向气球内吹气，吹气的方向始终垂直于墙面。
- 不考虑气体受压力影响的体积变化。
- 忽略钉子的长度，并且气球只要一接触墙面，就立刻发生爆炸，不管吹了多大。

特别的：

- 墙不一定是垂直于地面的

### Input

第一行包含一个整数  $L(1 \leq L \leq 10^3)$ ，表示 *Sumo* 每秒钟能匀速向气球中吹入  $L$  单位体积的气体。

第二行包含三个整数  $X, Y, Z(-10^3 \leq X, Y, Z \leq 10^3)$ ，表示 *Sumo* 嘴巴所在的位置。

接下来包含三行，每行包含三个整数  $x_i, y_i, z_i(-10^3 \leq x_i, y_i, z_i \leq 10^3)$ ，表示墙面上的三个点，保证三个点不共线。

### Output

输出仅包含一个实数，表示从 *Sumo* 往气球里吹起开始计时，第几秒气球发生爆炸。（结果与正确答案相差在  $10^{-7}$  以内都算正确，**末尾请勿输出回车**）

### Examples

Sample Input	Sample Output
10 0 10 0 0 0 0 0 0 10 10 0 0	52.3598775598

## Problem K. Sumo and Fibonacci

Time limit: 2 seconds  
Memory limit: 512 megabytes

众所周知 *Sumo* 非常擅长数学题，今天他碰到了一道和 Fibonacci（斐波那契）数列有关的题目。

*Sumo* 得到了一个长度为  $n$  的数组  $a$ 。

现在给出  $q$  次询问，每次询问给出一对正整数  $l, r$ ，将数组  $a$  中  $[l, r]$  区间的数  $a_l, a_{l+1}, \dots, a_r$  **排序并去重** 后得到一个长度为  $m$  的数组  $b$ 。现在给出 Fibonacci（斐波那契）数列的定义如下：

$$\begin{cases} Fib_1 = Fib_2 = 1 \\ Fib_i = Fib_{i-1} + Fib_{i-2} \quad (i \geq 3) \end{cases}$$

*Sumo* 需要你来帮助他求出  $\sum_{i=1}^m b_i \times Fib_i$  对  $10^9 + 7$  取模后的结果。

### Input

第一行给出一个正整数  $n(2 \leq n \leq 3 \times 10^4)$ ，表示数组  $a$  的长度。

第二行给出  $n$  个正整数，第  $i$  个数表示  $a_i(0 \leq a_i \leq 10^9)$ 。

第三行给出一个正整数  $q(2 \leq q \leq 3 \times 10^4)$ ，表示询问个数。

接下来  $q$  行每行一对正整数  $l, r(1 \leq l \leq r \leq n)$ ，表示询问的区间。

### Output

输出  $q$  行，第  $i$  行对应第  $i$  次询问给出的  $l, r$  的答案。

### Examples

Sample Input	Sample Output
5	9
1 2 3 4 5	21
3	28
1 3	
1 4	
2 5	

### Explanation

询问 1 中  $[1,3]$  区间排序去重后的结果为  $\{1,2,3\}$ ，计算过程为： $1 * 1 + 2 * 1 + 3 * 2 = 9$ ，

询问 2 中  $[1,4]$  区间排序去重后的结果为  $\{1,2,3,4\}$ ，计算过程为： $1 * 1 + 2 * 1 + 3 * 2 + 4 * 3 = 21$ ，

询问 3 中  $[2,5]$  区间排序去重后的结果为  $\{2,3,4,5\}$ ，计算过程为： $2 * 1 + 3 * 1 + 4 * 2 + 5 * 3 = 28$ 。

## Problem L. Sumo and Coins

Time limit: 1 second  
Memory limit: 512 megabytes

There are  $n$  coins on *Sumo*'s table, each coin has two sides,  $a$  ( $0 \leq a \leq n$ ) of the coins face up, and  $b$  ( $0 \leq b \leq n$ ) of the coins face down. *Sumo* wants to make these coins have the same side.

Each time he can flip any  $n - 1$  of these coins, he can do this operation any number of times (possibly zero).

Please tell him if he can make these coins have the same side.

### Input

The first line of the input is a single integer  $T$  ( $1 \leq T \leq 10^4$ ),  $T$  test cases follow.

Each test case has three integers  $n$  ( $1 \leq n \leq 10^4$ ),  $a$ ,  $b$  ( $a + b = n$ ).

### Output

If all the coins can face up or down, output "ALL" (without quotes);

If all the coins can only face up, output "UP" (without quotes);

If all the coins can only face down, output "DOWN" (without quotes);

If all the coins cannot face the same side, output "NULL" (without quotes).

### Examples

Sample Input	Sample Output
1 2 1 1	ALL

### Explanation

In the example, *Sumo* can flip the coin facing up or down so that all the coins are facing the same way after the first operation.