2017 ACM-ICPC Asia Taiwan Online Programming Contest Contest Information

Rules

Contestants will be disqualified if they violate any one of the following rules.

1. No machine-readable materials (e.g., source codes, templates, etc.) are allowed. However, paper-based materials, such as textbooks, dictionaries, printed notes, etc., are allowed.
2. Contestants are only allowed to contact his/her teammates during the contest. Contestants shall not discuss with his/her coach and other teams.
3. Contestants shall only access the internet for downloading the problem description, submitting source codes, requesting problem clarification and checking the scoreboard. Any other type of internet access is prohibited.
4. A team shall not simultaneously use more than one computer to write programs during the contest. Contestant shall not use any other type of electronic devices, except extra monitors and printers.
5. All malicious actions interfering the contest are prohibited.

Scoring and Ranking

1. Disqualified teams will be removed from the ranking.
2. Only C, C++, Java, Python are provided in this contest. The judge system only accepts programs which can be properly compiled and executed. The memory, time, and output limits of compilation are 3.5 gigabytes, 20 seconds, and 1000 megabytes, respectively. A problem is solved if the submitted program terminates and outputs correctly in time. The responses of the judge system are listed as follows.
   - Compilation Error: The program cannot be properly compiled or executed.
   - Time-Limit Exceeded: The program uses too much time.
   - Run-Time Error: The program cannot terminate normally.
   - Wrong Answer: The output is incorrect.
   - Yes: The program is accepted by the judge system, and the problem is solved.
3. Teams are ranked according to the most problems solved. Teams who solve the same number of problems are ranked by least total time. The total time is the sum of the time consumed for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the accepted run plus 20 penalty minutes for every rejected run for that problem regardless of submittal time. There is no time consumed for a problem that is not solved.
4. For any two teams have the same problems solved and the same penalty, the winner is determined by their first accepted submissions to problems solved by them. The loser goes to the team submitting the run of the greatest ID among these submissions.

Problem A
Similarity Computation

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

The Jaccard similarity coefficient is usually used for measuring the similarity of two sets. Give two sets A and B, the Jaccard similarity coefficient, $J(A, B)$, is defined as the size of the intersection divided by the size of the union of the two sets. That is, $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$. For example, if $A = \{1, 3, 7, 8\}$ and $B = \{1, 7, 9\}$, then $J(A, B) = \frac{|\{1,7\}|}{|\{1,3,7,8,9\}|} = \frac{2}{5}$.

Assume the element i in the set is an integer between 0 to 9 ($0 \leq i \leq 9$) and the size of the set is no larger than 10. Please write a program to compute the Jaccard similarity coefficient of two sets A and B. And output 1 if $J(A, B) > 0.5$ and 0 if $J(A, B) \leq 0.5$.

Input Format

The first line of the input file contains an integer T ($T \leq 25$) indicating the number of test cases to follow.

Each test case will consist of three lines. The first line contains two integers m and n ($0 < m, n \leq 10$), indicating the number of elements of sets A and B, respectively. The second line contains m integers (the elements of set A) and the third line contains n integers (the elements of set B).

You may assume:

- $1 \leq T \leq 25$
- $m \leq 10$ and $n \leq 10$

Output Format

For each test case, output 1 if $J(A, B) > 0.5$ and 0 if $J(A, B) \leq 0.5$.

Sample Input

```
3
5 6
0 2 3 5 6
1 2 4 6 7 9
3 2
1 4 6
4 6
7 7
0 1 3 4 6 8 9
0 1 2 3 4 6 7
```

Sample Output

```
0
1
1
```

Problem B
The Combination of Poker Cards

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

Poker is a popular card game worldwide, which was played from the 18th century in a variety of forms. A standard deck of poker cards has 52 cards divided into four suits, with each suit having the 13 ranks. In the Chinese area, "Big Two" and "Thirteen Cards" are two popular poker games based on the ranking of different combinations of cards. If we want to design such computer poker games, we need to write a program that can recognize card combinations. Suppose that we simplify the requirement as follows: Without considering the types of suits but only concerning about the ranks of cards, try to determine the card combination from any six given cards. If we use integer numbers to represent card ranks, the possible card combinations are described below:

- single: six different numbers, e.g., 2 5 7 10 9 8
- one pair: one pair of equal numbers, e.g., 4 4 7 10 8 9
- two pairs: two pairs of equal numbers, e.g., 8 8 3 3 6 7
- three pairs: three pairs of equal numbers, e.g., 8 8 3 3 7 7
- one triple: three equal numbers, e.g., 2 2 2 7 5 6
- two triples: two suits of three equal numbers, e.g., 2 2 2 7 7 7
- tiki: four equal numbers, e.g., 5 5 5 5 9 8
- tiki pair: four equal numbers and another one pair, e.g., 5 5 5 5 9 9
- full house: three equal numbers and another one pair, e.g., 3 3 3 9 9 7

Suppose that we use integer numbers from 1 to 13 to represent the card ranks. Please write a program to determine the card combination from six input numbers.

Input Format

The first input line contains one integer number T, indicating the number of test cases. Each test case includes six integer numbers (with each number from 1 to 13) that are separated by a single white space.

You may assume:

- $1 \leq T \leq 25$
- In every test case, any number appears at most 4 times.

Output Format

Show the name of the card combination for each input test case using the namespace of single, one pair, two pairs, three pairs, one triple, two triples, tiki, tiki pair, and full house.

Sample Input

```
5
4 3 4 3 12 10
5 4 2 3 6 12
10 12 10 12 12 8
8 5 8 8 5 5
```

```
2 10 6 10 10 10
```

Sample Output

```
two pairs
single
full house
two triples
tiki
```

Problem C
Coloring Intervals

Time limit: 3 seconds
Memory limit: 512 megabytes

Problem Description

For all real numbers $a \leq b$, the closed interval $[a, b]$ refers to the set of all real numbers between a and b, inclusive. For example, $[3, 5] = \{x \in \mathbb{R} \mid 3 \leq x \leq 5\}$.

Bob has n closed intervals, denoted $[a_1, b_1]$, $[a_2, b_2]$, ..., $[a_n, b_n]$, such that for all distinct i, $j \in \{1, 2, \ldots, n\}$,

- a_i and b_i are positive integers.
- $a_i \leq b_i$. I.e., $[a_i, b_i]$ is not empty.
- $a_i \neq b_j$, $a_i \neq a_j$ and $b_i \neq b_j$. I.e., distinct closed intervals do not have any common endpoint.

He wants to color each of the n closed intervals monochromatically such that any two distinct overlapping intervals are in different colors. Bob wonders how many colors are needed. In other words, he wants to find the minimum positive integer k such that each of the n closed intervals can be labelled with one of 1, 2, ..., k in a way that any two distinct overlapping intervals are labelled differently. Please help him.

Input Format

The first line contains the number T of test cases. Each of the next T lines specifies a test case by providing n, a_1, b_1, a_2, b_2, ..., a_n, b_n, in that order. Two consecutive numbers in a line are separated by one or more spaces.

You may assume:

- $1 \leq T \leq 10$
- $n \in \{2, 3, \ldots, 100000\}$
- a_i and b_i are positive integers less than or equal to $2^{32} - 1$ for each $i \in \{1, 2, \ldots, n\}$.

Output Format

For each test case, output the minimum positive integer k such that each of $[a_1, b_1]$, $[a_2, b_2]$, ..., $[a_n, b_n]$ can be labelled with one of 1, 2, ..., k in a way that any two distinct overlapping intervals are labelled differently.

Sample Input

```
3
4 1 2 3 5 4 8 6 7
5 2 5 1 10 3 7 4 6 8 9
4 3 7 4 5 2 9 6 8
```

Sample Output

```
2
4
3
```

Problem D
Mixing Coins

Time limit: 5 seconds
Memory limit: 512 megabytes

Problem Description

Misaka likes to shoot coins as a powerful railgun.

She prepares a line of coins to fight crime. To produce a stronger coin, she mixes coins together. However, coins with different materials are not compatible to each other, so she only mixes coins with same material together.

Here's the steps Misaka makes coins:

1. Find first three consecutive coins with same material from the beginning of the line
2. Take them out from the line
3. Mix together and produce a new coin with same material
4. Put the new coin at the end of line

She repeatedly do these steps until she can't produce new coins anymore.

Misaka wants to know how many coins she will have. Please help her count coins rapidly!

Input Format

On the first line there is a single integer T indicating the number of test cases.

The first line of each test case contains an integer N indicating the number of groups of consecutive coins Misaka has. All coins are in a single line.

Then N lines follow, each line containing a character c_i and an integer n_i, denoting that there are n_i consecutive coins with material c_i for i-th group of consecutive coins, behind $(i-1)$-th.

You may assume:

- $T \leq 10$
- $1 \leq N \leq 10^5$
- $1 \leq n_i \leq 10^9$
- c_i is an uppercase alphabet, $c_i \neq c_{i+1}$ for $1 \leq i < N$

Output Format

For each test case, output an integer in one line, indicating the number of coins after Misaka doing the steps of making coins as many as possible.

Sample Input

```
2
3
A 3
B 1
A 2
3
A 2
B 3
A 2
```

Sample Output

```
2
3
```

Problem E
Fences

Time limit: 3 seconds
Memory limit: 512 megabytes

Problem Description

Your friend, Donald, has a villa surrounded by two tiers of fences, and he wants to calculate the area of land between them. He can measure the length of any fence, but Donald has no idea on calculating the area. Watson, one of Donald's friends, notices that the fences are probably built by a computer scientist mastering the knowledge of computational geometry, because the following facts are no coincidence.

- The shape of the land inside the outer tier is a perfect circle C. Let B denote the set of points on the boundary of C.
- The shape of the land inside the inner tier is a non-self-intersecting polygon P of n vertices. I.e., two edges do not intersect if they don't share a common vertex. Let V denote the set of vertices of P.
- All vertices of P have identical minimum distances to C. In other words, for distinct vertices $(x_u, y_u), (x_v, y_v) \in V$, we have

$$\min_{(x,y)\in B} \sqrt{(x-x_u)^2+(y-y_u)^2} = \min_{(x,y)\in B} \sqrt{(x-x_v)^2+(y-y_v)^2}.$$

Suddenly, you know how to calculate the area of land between the two tiers of fences from the total length c of outer tier and the lengths $\ell_1, \ldots, \ell_n$ of the n edges of P. Note that Donald can measure these length. Could you help him to calculate the area?

Input Format

The first line of the input contains a positive integer T indicating the number of test cases. Each test case consists of two lines. The first line contains two numbers c and n separated by a space. c is the total length of the outer tier, i.e., c is the perimeter of C. n is the number of vertices of P. The second line contains n positive integers $\ell_1, \ldots, \ell_n$ indicating the lengths of edges of P.

You may assume:

- $1 \le T \le 100$
- $3 \le n \le 10$
- $10 \le c \le 1000$
- $\ell_1, \ldots, \ell_n > 0$
- P must be inside the circle C.

Output Format

For each case, output the area between the two tiers of fences. Your answer will be accepted if the absolute error or the relative error is less than 10^{-6}.

Sample Input

```
2
10.0 3
1 1 1
10.0 4
1 1 1 1
```

Sample Output

```
7.524734452702549
6.9577471545947684
```

Problem F
A Simple Function

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

A function $f : \mathbb{N}^3 \to \mathbb{N}$ where $\mathbb{N}$ stands for the set of non-negative integers, is defined as follows.

- $f(i, 0, M) = 1$, for all i and M.
- $f(i, i, M) = 1$, for all i and M.
- $f(i, x, M) = 0$, for all $i < x$.
- $f(i, x, M) = f(i-1, x-1, M) + f(i-1, x, M)$, if $f(i-1, x-1, M) + f(i-1, x, M)$ is NOT a multiple of M, for all $0 < x < i$.
- $f(i, x, M) = 0$, if $f(i-1, x-1, M) + f(i-1, x, M)$ is a multiple of M, for all $0 < x < i$.

For example, $f(2, 1, 2) = 0$ and $f(4, 2, 5) = 6$.

Input Format

The first line of the input contains an integer T, the number of test cases. T lines follow, one line per test case consisting of three space-separated integers a, b and M indicating that the value of $f(a, b, M)$ is to be computed.

You may assume:

- $1 \le T \le 10^4$
- $0 \le a < 2^{31}$
- $0 \le b < 2^{31}$
- $M \le 10000$ is a prime

Output Format

For each test case, output a single integer which denotes your answer modulo $10^9 + 7$ in a line.

Sample Input

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

Sample Output

```
0
6
```

Problem G
The Jet-Black Wings

Time limit: 3 seconds
Memory limit: 512 megabytes

Problem Description

"AHHHHHHHHH..."

Eddy, who calls himself "The Jet-Black Wings", is fighting against an evil organization called Dark Reunion. Then, he startled from the dream.

"I must be more powerful." Eddy said to himself in his mind.

Eddy often practice to be a powerful fighter. During his practice, he collects N magic stones. The i-th stone contains A_i units of dark forces. Eddy does the instruction for Q turns, each turn he has two choices:

1 X: Use X units of dark forces to all of the magic stones. Thus, the dark forces of the i-th magic stone changes to $A_i \oplus X$.

2 K: Sort all the magic stones by their dark forces increasely and sum up the dark forces of the first K magic stones.

Could you help Eddy to check whether he is correct?

Expression $x \oplus y$ means applying bitwise exclusive or operation to integers x and y. The given operation exists in all modern programming languages, for example, in languages C++ and Java it is represented as "^", in Pascal —as "xor".

Input Format

On the first line there is a single integer T indicating the number of test cases.

The first line of each test case contains two integers N, Q, indicating the number of magic stones and the number of instructions.

The second line of each test case contains N integers $A_1, A_2, \ldots, A_N$, indicating the dark forces of the i-th magic stone.

For the following Q lines, each line contains an instruction "1 X" or "2 K".

You may assume:

- $T \leq 1000$
- $1 \leq N, Q \leq 100000$
- $0 \leq A_i, X < 2^{31}$
- $1 \leq K \leq N$
- There are at most 5 test cases with $N + Q > 200$.

Output Format

For each "2 K" instruction, sum up the dark forces of the first K magic stones after sorted and output in one line.

Sample Input

```
1
3 6
4 8 3
1 3
```

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

Sample Output

```
17
7
3
```

Problem H
HH Country

Time limit: 10 seconds
Memory limit: 512 megabytes

Problem Description

HH is the world's leading competitive programming country. It consists of n cities labeled from 1 to n, and the cities are connected by roads. Under the planning of the shrewd king of HH, there is exactly one path between any two different cities. In other words, the cities of HH made up a clever tree structure.

HH launched a series of contests to arrange the budget of "Forward-looking Infrastructure Development Program". It has m rounds, and the i-th round will determine the distribution of the i-th budget. The distribution is based on the result of a double round-robin tournament among k_i cities associated with i-th budget. More specifically, considering any two different participating cities A and B. There will be a game that A send a team to the city B, as well as a game that B send a team to the city A. So there will be $k_i \times (k_i - 1)$ games in a round totally.

In order to be able to detail the travel expenses, HH country would like to ask you to calculate the total travelling distance between participating cities of each round. The distance of one game is defined as the number of roads on the only path from the away city to the home city.

Input Format

The first line contains an integer T, denoting the number of test cases. The first line of each test case contains two integers n and m, denoting the number of cities and rounds. Each of the next $n - 1$ lines contains two integers u and v, denoting a road between the city u and the city v. Each of the next m lines contains an integer k_i at the beginning, denoting the number of associated cities in i-th round; and followed by k_i integers $c_{i,j}$ in the same line denote the labeles of the these cities.

You may assume:

- $1 \leq T \leq 100$
- $2 \leq n \leq 10^5$
- $1 \leq u, v, c_{i,j} \leq n$
- $1 \leq m \leq 10^5$
- $2 \leq k_i \leq n$
- All $c_{i,j}$ are distinct in a round
- $\sum_i k_i \leq 2 \times 10^5$ in a test case
- The size of an input file is not greater than 60MB

Output Format

For each round of the contest, please output an integer, denoting the total travelling distance of that round.

Sample Input

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

Sample Output

```
2
2
6
16
36
```

2017 ACM-ICPC Asia Taiwan Online Programming Contest
競賽資訊

競賽規則

違反下列規則，將導致參賽者失去參賽資格。

一、 不得使用任何機器可讀的資料，如預先寫好存放於電腦中的程式碼。但可以使用紙本資料，如教科書、字典、筆記以及列印好的紙本程式碼。

二、 在比賽過程中，參賽者只能與隊友討論。競賽期間與教練或其他隊伍聯繫均屬違規行為。

三、 參賽者只能夠透過網路下載題目敘述、上傳解答程式碼、提問澄清疑點與查看計分板。使用網路存取其他資訊均屬違規。

四、 每個隊伍僅可使用一台電腦撰寫程式與上傳程式碼。於競賽期間除使用印表機列印題目與程式碼以及透過額外的螢幕閱讀題目之外，不得使用任何其他電子裝置。

五、 不得做出任何意圖妨礙比賽進行及影響比賽公平性的惡意行為。

計分與排名

一、 違反競賽規則以致失去參賽資格者，不予計分與排名。

二、 本次競賽僅提供 C、C++、Java、Python ，不接受其他程式語言。程式需能正常編譯執行，編譯時可使用的記憶體、時間、輸出上限分別為 3.5 gigabytes、20 秒、1000 megabytes。程式在時間限制內計算完畢輸出正確答案，才算答對。以下為裁判系統常見回應：

- Compilation Error: 語法有錯誤或其他因素以致於無法編譯或執行。
- Time-Limit Exceeded: 未能在時間限制內執行完畢。
- Run-Time Error: 執行時錯誤，程式無法正常執行完畢。
- Wrong Answer: 在時限內執行完畢但輸出錯誤。
- Yes: 答對。

三、 隊伍以解題數量多者排名較前，解題數量相同時，以總消耗時間少者排名較前。答對的題目的消耗時間計算方式為比賽開始至解出題目所消耗的分鐘數。如解出前有答錯，每答錯一次需要另加 20 分鐘。總消耗時間為所有答對題目的消耗時間加總。未答對的題目不計消耗時間。

四、 如果在前述計分規則任兩隊無法分出勝負時，則依據兩隊每題第一份答對的程式碼之 Run ID 決勝：Run ID 最大值較大者負。

Problem A
Similarity Computation

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

Jaccard similarity coefficient 是一種常用於度量集合相似度的指標，給定 A、B 兩個集合，其 Jaccard similarity coefficient 度量方法如下:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

其中分子代表的是兩個集合的交集元素個數，分母則是兩個集合聯集元素個數。例如 $A = \{1,3,7,8\}$、$B = \{1,7,9\}$，則 $J(A,B) = \frac{|\{1,7\}|}{|\{1,3,7,8,9\}|} = \frac{2}{5}$。假設集合的元素只會是整數數字 0 至 9，請寫一個程式判斷兩個集合 (集合大小皆不大於 10) 的 Jaccard similarity coefficient 是否大於 0.5，當大於 0.5 時輸出 1，反之輸出 0。

Input Format

測試輸入的第一行為一個整數 T，代表有多少組測試資料。每組測試資料有三行，其中第一行為兩個正整數 m 跟 n，分別代表 A 集合與 B 集合的元素個數。接下來的兩行為一系列 0 到 9 的整數，代表集合 A 與 B 內的元素，每一行的數字間用單一空格區隔。

可假設：

- $1 \leq T \leq 25$
- $m \leq 10$ and $n \leq 10$

Output Format

針對每一組測試資料，當 $J(A,B) > 0.5$ 時輸出 1，反之輸出 0。

Sample Input

```
3
5 6
0 2 3 5 6
1 2 4 6 7 9
3 2
1 4 6
4 6
7 7
0 1 3 4 6 8 9
0 1 2 3 4 6 7
```

Sample Output

```
0
1
1
```

Problem B
The Combination of Poker Cards

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

撲克牌是一種國際盛行的卡牌遊戲，有很多種玩法，據傳最早源自 18 世紀。標準撲克牌共 52 張牌，包含四種花色，每一花色共 13 張不同點數的牌。在華人地區盛行的撲克牌玩法有大老二和十三支，主要是比較牌型組合的大小。如果想要利用電腦設計這些撲克牌遊戲，就必須寫程式自動判斷各種牌型。假設我們簡化牌型的判斷條件如下: 在不看花色與只看牌點的條件下，每次從給定的 6 張牌中，決定牌型是哪一種。如果使用整數數字代表不同的牌點，則可能的牌型描述如下:

- 單張 (single): 6 個數字均不相同，例如，2 5 7 10 9 8
- 一對 (one pair): 兩個數字相同，其餘均不同，例如，4 4 7 10 8 9
- 兩對 (two pairs): 兩組兩個數字相同，例如，8 8 3 3 6 7
- 三對 (three pairs): 三組兩個數字相同，例如，8 8 3 3 7 7
- 三條 (one triple): 一組三個數字相同，其餘均不同，例如，2 2 2 7 5 6
- 雙三條 (two triples): 兩組三個數字相同，例如，2 2 2 7 7 7
- 鐵支 (tiki): 四個數字相同，例如，5 5 5 5 9 8
- 鐵支配對 (tiki pair): 四個數字相同，另外還有一對數字，例如，5 5 5 5 9 9
- 葫蘆 (full house): 三個數字相同，另外還有一對數字，例如，3 3 3 9 9 7

假設使用整數數字 1 至 13 代表不同點數的牌卡，請寫一個程式從每行輸入的 6 個數字中判讀是以上哪一種牌型。

Input Format

第一行輸入包含一個整數 T，表示測試的牌型數量。每一個測試牌型包括 6 個落於 1 至 13 的整數數字，數字間用單一空格區隔。

可假設：

- $1 \leq T \leq 25$
- 不會有同一數字出現超過四次。

Output Format

輸出每一行的牌型名稱，牌型名稱請用對應的英文名稱，即 single、one pair、two pairs、three pairs、one triple、two triples、tiki、tiki pair 或 full house。

Sample Input

```
5
4 3 4 3 12 10
5 4 2 3 6 12
10 12 10 12 12 8
8 5 8 8 5 5
2 10 6 10 10 10
```

Sample Output

```
two pairs
single
full house
two triples
tiki
```

Problem C
Coloring Intervals

Time limit: 3 seconds
Memory limit: 512 megabytes

Problem Description

對任兩個實數 $a \le b$，閉區間 $[a, b]$ 乃指介於 a 與 b 之間（包含 a 與 b）的實數之集合。例如，$[3, 5] = \{x \in \mathbb{R} \mid 3 \le x \le 5\}$。

鮑伯有 n 個閉區間，分別為 $[a_1, b_1]$、$[a_2, b_2]$、$\ldots$、$[a_n, b_n]$，其中對所有相異 i、$j \in \{1, 2, \ldots, n\}$，

- a_i, b_i 均為正整數
- $a_i \le b_i$，即 $[a_i, b_i]$ 不是空集合。
- $a_i \ne b_j$、$a_i \ne a_j$ 且 $b_i \ne b_j$，即相異區間不會有共通的端點。

他想要將每個閉區間塗上一個顏色，且使任兩相異且有交集的區間被塗上相異的顏色。鮑伯想知道最少要有幾個顏色才可以做這件事。換句話說，他想找到最小的正整數 k，使得 n 個閉區間中的都可以被賦予 1、2、$\ldots$、k 中的一個值，而使任兩相異且相交的區間被賦予的值相異。請幫他算吧。

Input Format

第一行給出測資筆數 T。接下來的 T 行中，每一行都指定一筆測資，指定的方法是依序給出 n、a_1、b_1、a_2、b_2、$\ldots$、a_n、b_n。同一行中的連續兩數字將被一或多個空白隔開。

可假設：

- $1 \le T \le 10$
- $n \in \{2, 3, \ldots, 100000\}$
- 對每個 $i \in \{1, 2, \ldots, n\}$，$a_i$ 與 b_i 都是小於等於 $2^{32} - 1$ 的正整數。

Output Format

對每一筆測資，輸出最小的正整數 k，使得 $[a_1, b_1]$、$[a_2, b_2]$、$\ldots$、$[a_n, b_n]$ 中的任何一個都可以被賦予 1、2、$\ldots$、k 中的一個值，而使任兩相異且相交的區間被賦予的值相異。

Sample Input

```
3
4 1 2 3 5 4 8 6 7
5 2 5 1 10 3 7 4 6 8 9
4 3 7 4 5 2 9 6 8
```

Sample Output

```
2
4
3
```

Problem D
Mixing Coins

Time limit: 5 seconds
Memory limit: 512 megabytes

Problem Description

御坂喜歡用硬幣當作電磁炮射擊。

為了打擊犯罪，她準備了一排硬幣。為了生產出更強力的硬幣，她會把硬幣們混合在一起。然而，不同材質的硬幣之間並不相容，所以她只會將相同材質的硬幣混合。

以下是御坂製作硬幣的步驟：

1. 從序列的頭開始，找出第一組連續三個相同材質的硬幣
2. 將它們從序列取出
3. 混合在一起，生產出一枚新的相同材質的硬幣
4. 將新硬幣放回序列尾端

她會重複做這些步驟，直到她不能再生產新的硬幣。

御坂想要知道她最後會有多少硬幣。請趕快幫她計算硬幣吧！

Input Format

第一行有一個整數 T ，表示測試資料的數量。

每組測試資料的第一行有一個正整數 N，表示御坂有 N 組連續的硬幣。所有硬幣都在一排之中。

接著有 N 行，每一行有一個字元 c_i 和正整數 n_i ，表示第 i 組有連續 n_i 個材質為 c_i 的硬幣，接在第 $(i-1)$ 組硬幣之後。

- $T \leq 10$
- $1 \leq N \leq 10^5$
- $1 \leq n_i \leq 10^9$
- c_i 是一個英文大寫字母，$c_i \neq c_{i+1}$ 對於 $1 \leq i < N$

Output Format

對於每組測試資料，輸出一個整數於一行，表示御坂在做完盡量多次製作硬幣的步驟之後，有多少硬幣。

Sample Input

```
2
3
A 3
B 1
A 2
3
A 2
B 3
A 2
```

Sample Output

```
2
```

3

Problem E
Fences

Time limit: 3 seconds
Memory limit: 512 megabytes

Problem Description

你的朋友東納德的別墅外有兩層圍籬，他想要計算在這兩層圍籬之間的土地面積。他能測量圍籬的長度，但他對計算面積毫無概念。東納德的朋友華生發現圍籬可能是由專精於計算幾何的電腦科學家建造的，因為下列事實絕非巧合：

- 外層圍籬之內的土地，形成一個完美的圓形 C。以下用 B 代表 C 邊界上的所有點所形成的集合。
- 內層圍籬之內的土地，形成一個非自交 n 邊形 P，即不共用頂點的兩條邊，沒有交點。以下用 V 代表 P 的所有頂點所形成的集合。
- P 所有頂點到 C 的最短距離都是一樣的，即對 V 中相異的兩頂點 $(x_u, y_u), (x_v, y_v)$，下述等式成立：

$$\min_{(x,y)\in B} \sqrt{(x-x_u)^2+(y-y_u)^2} = \min_{(x,y)\in B} \sqrt{(x-x_v)^2+(y-y_v)^2}.$$

你恍然大悟。你明白了如何由外層圍籬總長 c 以及內層圍籬的 n 個邊長 $\ell_1, \ldots, \ell_n$ 計算出兩個圍籬之間的土地面積。還記得東納德知道如何測量圍籬長度吧？請幫他算出面積吧。

Input Format

輸入的第一行有一個正整數 T 代表有多少筆測試資料。每一筆測試資料有兩行，第一行有兩個數字 c 跟 n，以一個空白隔開。c 代表了外層圍籬的總長，也就是 C 的周長。n 代表內層圍籬的頂點數目。第二行有 n 個正整數 $\ell_1, \ldots, \ell_n$，代表 n 邊形 P 的各個邊長。

可假設

- $1 \le T \le 100$
- $3 \le n \le 10$
- $10 \le c \le 1000$
- $\ell_1, \ldots, \ell_n > 0$
- P 必在圓 C 內部。

Output Format

對每筆測試資料，請輸出兩層圍籬間的土地面積。只要絕對或相對誤差小於 10^{-6} 就會被視為正確。

Sample Input

```
2
10.0 3
1 1 1
10.0 4
1 1 1 1
```

Sample Output

```
7.524734452702549
6.9577471545947684
```

Problem F
A Simple Function

Time limit: 1 second
Memory limit: 512 megabytes

Problem Description

定義一個函數 $f : \mathbb{N}^3 \to \mathbb{N}$，其中 $\mathbb{N}$ 代表非負整數集合，f 的定義如下：

- 對於所有的 i 和 M，$f(i, 0, M) = 1$
- 對於所有的 i 和 M，$f(i, i, M) = 1$
- 對於所有的 $i < x$，$f(i, x, M) = 0$
- 對於所有的 $0 < x < i$，若 $f(i-1, x-1, M) + f(i-1, x, M)$ 不是 M 的倍數，則 $f(i, x, M) = f(i-1, x-1, M) + f(i-1, x, M)$
- 對於所有的 $0 < x < i$，若 $f(i-1, x-1, M) + f(i-1, x, M)$ 是 M 的倍數，則 $f(i, x, M) = 0$

例如 $f(2, 1, 2) = 0$，$f(4, 2, 5) = 6$.

Input Format

第一行有一個整數 T 代表測試資料組數。每組測試資料由三個整數 a, b 和 M 組成，請計算 $f(a, b, M)$ 的值。

可假設：

- $1 \le T \le 10^4$
- $0 \le a < 2^{31}$
- $0 \le b < 2^{31}$
- $M \le 10000$ 且是個質數

Output Format

對於每組測試資料，請輸出一個整數，代表計算得到的答案除以 $10^9 + 7$ 的餘數。

Sample Input

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

Sample Output

```
0
6
```

Problem G
The Jet-Black Wings

Time limit: 3 seconds
Memory limit: 256 megabytes

Problem Description

「呃啊... 可惡！... 要發狂了嗎！」

艾迪，一個自稱為「漆黑之翼」的人，正在與名為「Dark Reunion」的邪惡組織作戰。接著他就驚醒了，原來只是一場夢阿...

「我一定要變得更強。」艾迪在心中激勵自己。

為了成為一名強悍的戰鬥者，艾迪認真的鍛鍊自己。在他的練習中，他收集了 N 顆魔法石頭，第 i 顆魔法石頭有著 A_i 單位的黑暗力量。艾迪會進行 Q 次操作，每次操作有以下兩種：

1 X: 使用 X 單位的黑暗力量於每顆魔法石頭. 因此，第 i 顆魔法石頭的暗黑力量會變成 $A_i \oplus X$ 單位。

2 K: 將所有的魔法石頭按照他們的黑暗力量由小到大排序，並加總前 K 顆魔法石頭的黑暗力量。

你能夠幫助艾迪確認他是否正確嗎？

$x \oplus y$ 表示將 x 與 y 進行互斥或操作。這個操作存在於所有常用的程式語言中，例如：C++ 與 Java 即是使用「^」，而 Pascal 則使用「xor」。

Input Format

第一行有一個數字 T，表示有 T 組測試資料。

每組測試資料的第一行有兩個數字 N, Q，表示艾迪蒐集的魔法石頭個數與訓練的操作次數。

每組測試資料的第二行有 N 個數字 $A_1, A_2, \ldots, A_N$，其中 A_i 表示第 i 顆魔法石頭的黑暗力量。

接著有 Q 行，每行為一個操作「1 X」或「2 K」。

可假設：

- $T \leq 1000$
- $1 \leq N, Q \leq 100000$
- $0 \leq A_i, X < 2^{31}$
- $1 \leq K \leq N$
- 至多只有 5 組測試資料的 $N + Q > 200$。

Output Format

對於每個操作「2 K」輸出一個數字於一行，表示排序後前 K 顆魔法石頭的黑暗力量總和。

Sample Input

```
1
3 6
4 8 3
1 3
1 1
2 3
1 2
2 2
```

```
2 1
```

Sample Output

```
17
7
3
```

Problem H
HH Country

Time limit: 10 seconds
Memory limit: 512 megabytes

Problem Description

HH 國是世界知名的競技程式設計比賽強國。其國土由 n 座編號 1 到 n 的城市組成，城市之間以道路連接。在精明的 HH 國王規劃之下，任兩座城市之間恰有一條經過若干道路的路徑可以往返。換言之，HH 國的城市組成了一個巧妙的樹型結構。

為了分配各城市在前瞻基礎建設計畫中的預算，HH 國展開了一系列的比賽。比賽共有 m 輪，第 i 輪比賽會決定第 i 項預算的分配，而分配方式則基於與此項預算相關的 k_i 個城市之間雙循環賽的結果。具體而言，任兩個相異的參賽城市 A, B 之間，會有一場由 A 派代表隊去城市 B 進行的比賽，以及一場由 B 派代表隊去城市 A 進行的比賽。整輪共會進行 $k_i \times (k_i - 1)$ 場。

為了能夠詳實的編列交通費，HH 國想請你幫忙算出每輪比賽中，參賽城市間的總移動距離。而其中一場比賽的移動距離則定義為客場城市到主場城市之唯一路徑上的道路數量。

Input Format

第一行有一個整數 T，代表有多少筆測試資料。每筆測試資料的第一行有兩個整數 n, m，分別代表 HH 國的城市數以及有幾輪比賽。接下來 $n - 1$ 每行有兩個整數 u, v，代表有一條道路連接城市 u 及城市 v。接下來 m 行，每行開頭有一個整數 k_i，代表有幾座城市參與該輪比賽；同一行中接下來 k_i 個整數 $c_{i,j}$ 則代表這 k_i 座城市的編號。

可假設：

- $1 \le T \le 100$
- $2 \le n \le 10^5$
- $1 \le u, v, c_{i,j} \le n$
- $1 \le m \le 10^5$
- $2 \le k_i \le n$
- 一輪比賽中所有 $c_{i,j}$ 皆相異
- 一筆測試資料中，$\sum_i k_i \le 2 \times 10^5$
- 一個輸入檔之大小不超過 60MB

Output Format

對於每一輪比賽，請輸出一個整數，代表該輪比賽中，參賽城市間的總移動距離。

Sample Input

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

Sample Output

```
2
2
6
16
36
```