The 10th Shandong Provincial Collegiate Programming Contest

Contest Session

May 12, 2019

Problem List

A	Calandar
B	Flipping Game
C	Wandering Robot
D	Game on a Graph
E	BaoBao Loves Reading
F	Stones in the Bucket
G	Heap
H	Tokens on the Segments
I	Connected Intervals
J	Triangle City
K	Happy Equation
L	Median
M	Sekiro

This problem set should contain 13 (thirteen) problems on 16 (sixteen) numbered pages. Please inform a runner immediately if something is missing from your problem set.

Prepared by SUA Programming Contest Problem Setter Team.
https://sua.ac/

Hosted by

山东计算机学会

Shandong Computer Federation

Sponsored by

跃骏 YUEJUN

Chinesec 奇安信

SD·LeCroy

Problem Set Prepared by

It’s against the rules to open non-contest websites during the contest.
If you’re interested (which is our pleasure), please scan the
QR code only after the contest.

Problem A. Calandar

On a planet far away from Earth, one year is composed of 12 months, and each month always consists of 30 days.

Also on that planet, there are 5 days in a week, which are Monday, Tuesday, Wednesday, Thursday and Friday. That is to say, if today is Monday, then tomorrow will be Tuesday, the day after tomorrow will be Wednesday. After 3 days it will be Thursday, after 4 days it will be Friday, and after 5 days it will again be Monday.

Today is the d_1-th day in the m_1-th month of year y_1. Given the day of today on that planet, what day will it be (or was it) on the d_2-th day in the m_2-th month of year y_2 on that planet?

Input

There are multiple test cases. The first line of the input contains an integer T (about 100), indicating the number of test cases. For each test case:

The first line contains three integers y_1, m_1, d_1 ($2000 \le y_1 \le 10^9$, $1 \le m_1 \le 12$, $1 \le d_1 \le 30$) and a string s, indicating the date and day of today on that planet. It's guaranteed that s is either "Monday", "Tuesday", "Wednesday", "Thursday" or "Friday".

The second line contains three integers y_2, m_2 and d_2 ($2000 \le y_2 \le 10^9$, $1 \le m_2 \le 12$, $1 \le d_2 \le 30$), indicating the date whose day we want to know.

Output

For each test case output one line containing one string, indicating the day of the d_2-th day in the m_2-th month of year y_2 on that planet.

Example

standard input	standard output
4 2019 5 12 Monday 2019 5 14 2019 5 12 Tuesday 2019 12 30 2019 5 12 Friday 1000000000 1 1 1000000000 1 1 Wednesday 2019 5 12	Wednesday Friday Thursday Thursday

Problem B. Flipping Game

Little Sub loves playing the game *Flip Me Please*. In the game, n lights, numbered from 1 to n, are connected separately to n switches. The lights may be either on or off initially, and pressing the i-th switch will change the i-th light to its opposite status (that is to say, if the i-th light is on, it will be off after the i-th switch is pressed, and vice versa).

The game is composed of exactly k rounds, and in each round, the player must press exactly m different switches. The goal of the game is to change the lights into their target status when the game ends.

Little Sub has just come across a very hard challenge and he cannot solve it. As his friend, it's your responsibility to find out how many solutions there are to solve the challenge and tell him the answer modulo 998244353.

We consider two solutions to be different if there exist two integers i and j such that $1 \leq i \leq k$, $1 \leq j \leq n$ and the j-th switch is pressed during the i-th round of the first solution while it is not pressed during the i-th round of the second solution, or vice versa.

Input

There are multiple test cases. The first line of the input contains an integer T (about 1000), indicating the number of test cases. For each test case:

The first line contains three integers n, k, m ($1 \leq n, k \leq 100$, $1 \leq m \leq n$).

The second line contains a string s ($|s| = n$) consisting of only '0' and '1', indicating the initial status of the lights. If the i-th character is '1', the i-th light is initially on; If the i-th character is '0', the i-th light is initially off.

The third line contains a string t ($|t| = n$) consisting of only '0' and '1', indicating the target status of the lights. If the i-th character is '1', the i-th light must be on at the end of the game; If the i-th character is '0', the i-th light must be off at the end of the game.

It is guaranteed that there won't be more than 100 test cases that $n > 20$.

Output

For each test case output one line containing one integer, indicating the answer.

Example

standard input	standard output
3 3 2 1 001 100 3 1 2 001 100 3 3 2 001 100	2 1 7

Note

For the first sample test case, Little Sub can press the 1st switch in the 1st round and the 3rd switch in the 2nd round; Or he can press the 3rd switch in the 1st round and the 1st switch in the 2nd round. So the answer is 2.

For the second sample test case, Little Sub can only press the 1st and the 3rd switch in the 1st and only round. So the answer is 1.

Problem C. Wandering Robot

DreamGrid creates a programmable robot to explore an infinite two-dimension plane. The robot has a basic instruction sequence $a_1, a_2, \ldots a_n$ and a "repeating parameter" k, which together form the full instruction sequence $s_1, s_2, \ldots, s_n, s_{n+1}, \ldots, s_{nk}$ and control the robot.

There are 4 types of valid instructions in total, which are 'U' (up), 'D' (down), 'L' (left) and 'R' (right). Assuming that the robot is currently at (x, y), the instructions control the robot in the way below:

- U: Moves the robot to $(x, y + 1)$.
- D: Moves the robot to $(x, y - 1)$.
- L: Moves the robot to $(x - 1, y)$.
- R: Moves the robot to $(x + 1, y)$.

The full instruction sequence can be derived from the following equations

$$\begin{cases} s_i = a_i & \text{if } 1 \le i \le n \\ s_i = s_{i-n} & \text{otherwise} \end{cases}$$

The robot is initially at $(0, 0)$ and executes the instructions in the full instruction sequence one by one. To estimate the exploration procedure, DreamGrid would like to calculate the largest Manhattan distance between the robot and the start point $(0, 0)$ during the execution of the nk instructions.

Recall that the Manhattan distance between (x_1, y_1) and (x_2, y_2) is defined as $|x_1 - x_2| + |y_1 - y_2|$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains two integers n and k ($1 \le n \le 10^5, 1 \le k \le 10^9$), indicating the length of the basic instruction sequence and the repeating parameter.

The second line contains a string $A = a_1a_2 \ldots a_n$ ($|A| = n$, $a_i \in \{$'L', 'R', 'U', 'D'$\}$), where a_i indicates the i-th instruction in the basic instriction sequence.

It's guaranteed that the sum of $|A|$ of all test cases will not exceed 2×10^6.

Output

For each test case output one line containing one integer indicating the answer.

Example

standard input	standard output
2 3 3 RUL 1 1000000000 D	4 1000000000

Note

For the first sample test case, the final instruction sequence is "RULRULRUL" and the route of the robot is (0, 0) - (1, 0) - (1, 1) - (0, 1) - (1, 1) - (1, 2) - (0, 2) - (1, 2) - (1, 3) - (0, 3). It's obvious that the farthest point on the route is (1, 3) and the answer is 4.

Problem D. Game on a Graph

There are k people playing a game on a connected undirected simple graph with n ($n \geq 2$) vertices (numbered from 0 to $(n-1)$) and m edges. These k people, numbered from 0 to $(k-1)$, are divided into two groups and the game goes as follows:

- They take turns to make the move. That is to say, person number 0 will make the 1st move, person number 1 will make the 2nd move, ..., person number ($i \mod k$) will make the $(i+1)$-th move.
- During a move, the current player MUST select an edge from the current graph and remove it. If the graph is no longer connected after removing the edge, the group this person belongs to loses the game (and of course their opponents win), and the game ends immediately.

Given the initial graph when the game starts, if all people use the best strategy to win the game for their groups, which group will win the game?

Recall that a simple graph is a graph with no self loops or multiple edges.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer k ($2 \leq k \leq 10^5$), indicating the number of people.

The second line contains a string $s_0 s_1 \ldots s_{k-1}$ of length k ($s_i \in \{‘1’, ‘2’\}$). $s_i =$ ‘1’ indicates that person number i belongs to the 1st group, and $s_i =$ ‘2’ indicates that person number i belongs to the 2nd group.

The third line contains two integers n and m ($2 \leq n \leq 10^5$, $n-1 \leq m \leq 10^5$), indicating the number of vertices and edges of the initial graph.

The following m lines each contains two integers u_i and v_i ($0 \leq u_i, v_i < n$), indicating that there is an edge connecting vertex u_i and v_i in the initial graph.

It's guaranteed that:

- The initial graph is a connected undirected simple graph.
- There exist two people who belong to different groups.
- The sum of k, the sum of n and the sum of m in all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer. If the 1st group wins, output “1” (without quotes); If the 2nd group wins, output “2” (without quotes).

Example

standard input	standard output
3 5 11212 4 6 0 1 0 2 0 3 1 2 1 3 2 3 5 11121 5 7 0 2 1 3 2 4 0 3 1 2 3 2 4 1 3 121 4 3 0 1 0 2 1 3	2 1 2

Problem E. BaoBao Loves Reading

BaoBao is a good student who loves reading, but compared with his huge bookshelf containing lots and lots of books, his reading desk, which can only hold at most k books, is surprisingly small.

Today BaoBao decides to read some books for n minutes by the desk. According to his reading plan, during the i-th minute, he is scheduled to read book a_i. The reading desk is initially empty and all the books are initially on the shelf. If the book BaoBao decides to read is not on the desk, BaoBao will have to fetch it from the shelf. Also, if the desk is full and BaoBao has to fetch another book from the shelf, he will have to put one book back from the desk to the shelf before fetching the new book.

Tired of deciding which book to put back, BaoBao searches the Internet and discovers an algorithm called the *Least Recently Used* (LRU) algorithm. According to the algorithm, when BaoBao has to put a book back from the desk to the shelf, he should put back the least recently read book.

For example, let's consider the reading plan {4, 3, 4, 2, 3, 1, 4} and assume that the capacity of the desk is 3. The following table explains what BaoBao should do according to the LRU algorithm. Note that in the following table, we use a pair of integer (a, b) to represent a book, where a is the index of the book, and b is the last time when this book is read.

Minute	Books on the Desk Before This Minute	BaoBao's Action
1	{}	Fetch book 4 from the shelf
2	{(4, 1)}	Fetch book 3 from the shelf
3	{(4, 1), (3, 2)}	Do nothing as book 4 is already on the desk
4	{(4, 3), (3, 2)}	Fetch book 2 from the shelf
5	{(4, 3), (3, 2), (2, 4)}	Do nothing as book 3 is already on the desk
6	{(4, 3), (3, 5), (2, 4)}	Put book 4 back to the shelf as its the least recently read book, and fetch book 1 from the shelf
7	{(3, 5), (2, 4), (1, 6)}	Put book 2 back to the shelf as its the least recently read book, and fetch book 4 from the shelf

Given the reading plan, what's the number of times BaoBao fetches a book from the shelf if the value of k (the capacity of the desk) ranges from 1 to n (both inclusive)?

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the length of the reading plan.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le n$), indicating the indices of the books to read.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line containing n integers $f_1, f_2, \dots, f_n$ separated by a space, where f_i indicates the number of times BaoBao fetches a book from the shelf when the capacity of the desk is i.

Please, DO NOT output extra spaces at the end of each line, or your solution may be considered incorrect!

Example

standard input	standard output
1 7 4 3 4 2 3 1 4	7 6 5 4 4 4 4

Problem F. Stones in the Bucket

There are n buckets on the ground, where the i-th bucket contains a_i stones. Each time one can perform one of the following two operations:

- Remove a stone from one of the non-empty buckets.
- Move a stone from one of the buckets (must be non-empty) to any other bucket (can be empty).

What's the minimum number of times one needs to perform the operations to make all the buckets contain the same number of stones?

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the number of buckets.

The second line contains n integers $a_1, a_2, \dots, a_n$ ($0 \le a_i \le 10^9$), indicating the number of stones in the buckets.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line containing one integer, indicating the minimum number of times needed to make all the buckets contain the same number of stones.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>4
3
1 1 0
4
2 2 2 2
3
0 1 4
1
1000000000</td><td>2
0
3
0</td></tr>
</table>

Note

For the first sample test case, one can remove all the stones in the first two buckets.

For the second sample test case, as all the buckets have already contained the same number of stones, no operation is needed.

For the third sample test case, one can move 1 stone from the 3rd bucket to the 1st bucket and then remove 2 stones from the 3rd bucket.

Problem G. Heap

DreamGrid is learning the insertion operation of a heap in the data structure course.

In the following description, we denote $i/2$ to be the maximum integer x that $2x \le i$. Recall that

- A heap of size n is an array $a_1, a_2, \ldots, a_n$ which satisfies one of the following two conditions:
  - For all $2 \le i \le n$, $a_{i/2} \le a_i$. This is called a min heap.
  - For all $2 \le i \le n$, $a_{i/2} \ge a_i$. This is called a max heap.
- The insertion operation can be described by the following pseudo-code:

```
procedure insert(
  v: value to insert,
  a: heap array,
  is_max: if a is a max heap)
{Let n be the length of the heap array after insertion}
a_n := v
i := n
while i > 1
  if is_max is false and a_{i/2} ≤ a_i
    {The heap array now satisfies the condition to be a min heap}
    break
  else if is_max is true and a_{i/2} ≥ a_i
    {The heap array now satisfies the condition to be a max heap}
    break
  swap a_{i/2} and a_i
  i := i/2
{Insertion ends}
```

DreamGrid has prepared an initially empty array a as the heap array and n integers $v_1, v_2, \ldots, v_n$. He is just about to insert these n integers into the heap array in order when his cellphone rings, so he leaves this work to his roommate BaoBao.

Unfortunately, BaoBao doesn't understand what the argument is_max means in the insertion function (but for our dear contestants, we hope that you've understood the meaning of this argument), so he generates a binary string (a string which only contains '0' and '1') $b = b_1b_2\ldots b_n$ of length n, where b_i indicates the i-th character in the string, and decides the value of is_max according to the string. When inserting v_i into a, if b_i equals to '0', then is_max **during this insertion** will be false; otherwise if b_i equals to '1', then is_max **during this insertion** will be true.

When DreamGrid comes back, he finds with dismay that the final "heap" array $a_1, a_2 \ldots, a_n$ does not seem to be a valid heap! Given the n inserted integers $v_1, v_2, \ldots, v_n$, the final array and given that BaoBao has inserted $v_1, v_2, \ldots, v_n$ in order, please help DreamGrid restore the binary string b BaoBao generates.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 10^5$), indicating the size of the final array.

The second line contains n integers $v_1, v_2, \ldots, v_n$ ($1 \le v_i \le 10^9$), indicating the integers in the order they are inserted.

The third line contains n integers $a_1, a_2, \ldots, a_n$, which is a permutation of $v_1, v_2, \ldots, v_n$, indicating the final "heap" array.

It's guaranteed that the sum of n of all test cases will not exceed 10^6.

Output

For each test case output one line containing one binary string, indicating the string BaoBao generates for inserting the integers. If there are multiple valid answers, output the one with the smallest lexicographic order. If the binary string does not exist, output "Impossible" (without quotes) instead.

Recall that, for two binary strings s and t of length n, we say s is lexicographically smaller than t, if there exists an integer k satisfying all the following constraints:

- $1 \le k \le n$.
- For all $1 \le i < k$, $s_i = t_i$.
- $s_k =$ '0' and $t_k =$ '1'.

Example

standard input	standard output
3 4 2 3 1 4 4 1 3 2 5 4 5 1 2 3 3 4 1 5 2 3 1 1 2 2 1 1	0101 Impossible 001

Note

We now explain the first sample test case.

i	v_i	b_i	**"Heap" Array after Insertion**
1	2	0	{2}
2	3	1	{3, 2}
3	1	0	{1, 2, 3}
4	4	1	{4, 1, 3, 2}

Problem H. Tokens on the Segments

Consider n segments on a two-dimensional plane, where the endpoints of the i-th segment are (l_i, i) and (r_i, i). One can put as many tokens as he likes on the integer points of the plane (recall that an integer point is a point whose x and y coordinates are both integers), but the x coordinates of the tokens must be different from each other.

What's the maximum possible number of segments that have at least one token on each of them?

Input

The first line of the input contains an integer T (about 100), indicating the number of test cases. For each test case:

The first line contains one integer n ($1 \le n \le 10^5$), indicating the number of segments.

For the next n lines, the i-th line contains 2 integers l_i, r_i ($1 \le l_i \le r_i \le 10^9$), indicating the x coordinates of the two endpoints of the i-th segment.

It's guaranteed that at most 5 test cases have $n \ge 100$.

Output

For each test case output one line containing one integer, indicating the maximum possible number of segments that have at least one token on each of them.

Example

standard input	standard output
2 3 1 2 1 1 2 3 3 1 2 1 1 2 2	3 2

Note

For the first sample test case, one can put three tokens separately on (1, 2), (2, 1) and (3, 3).

For the second sample test case, one can put two tokens separately on (1, 2) and (2, 3).

Problem I. Connected Intervals

DreamGrid has just found a tree of n vertices in his backyard. As DreamGrid loves connected components, he defines an interval $[l, r]$ ($1 \le l \le r \le n$) as a "connected interval" if the induced subgraph formed from the set $\mathbb{V} = \{v_i | i \in [l, r]\}$ consists of exactly one connected component, where v_i indicates the vertex whose index is i.

Given the tree in DreamGrid's backyard, your task is to help DreamGrid count the number of connected intervals.

Recall that an induced subgraph G' of a graph G is another graph, formed from a subset $\mathbb{V}$ of the vertices of the graph G and all of the edges in graph G connecting pairs of vertices in $\mathbb{V}$.

Input

There are multiple test cases. The first line of the input contains an integer T indicating the number of test cases. For each test case:

The first line contains an integer n ($1 \le n \le 3 \times 10^5$) indicating the size of the tree.

For the following $(n-1)$ lines, the i-th line contains two integers a_i and b_i ($1 \le a_i, b_i \le n$) indicating that there is an edge connecting vertex a_i and vertex b_i in the tree.

It's guaranteed that the given graph is a tree and that the sum of n in all test cases will not exceed 3×10^5.

Output

For each test case output one line containing one integer, indicating the number of connected intervals.

Example

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td><pre>2
4
1 2
2 3
3 4
4
1 2
2 3
2 4</pre></td><td><pre>10
9</pre></td></tr>
</table>

Note

For the first sample test case, all intervals are connected intervals.

For the second sample test case, all intervals but [3, 4] are connected intervals.

Problem J. Triangle City

Triangle City is a city with $\frac{n(n+1)}{2}$ intersections arranged into n rows and n columns, where the i-th row contains i intersections.

The intersections are connected by bidirectional roads. Formally, if we denote (i, j) as the intersection on the i-th row and the j-th column, for all $1 \le j \le i < n$,

- there is a road whose length is $a_{i,j}$ connecting intersection (i, j) and $(i + 1, j)$, and
- there is a road whose length is $b_{i,j}$ connecting intersection (i, j) and $(i + 1, j + 1)$, and
- there is a road whose length is $c_{i,j}$ connecting intersection $(i + 1, j)$ and $(i + 1, j + 1)$.

What's more, for all $1 \le j \le i < n$, there exists a triangle whose sides are of length $a_{i,j}$, $b_{i,j}$ and $c_{i,j}$. That's why the city is called the Triangle City!

Our famous traveler BaoBao has just arrived in the Triangle City, planning to start his journey from intersection $(1, 1)$ and end his trip at intersection (n, n). To fully enjoy the landscape, BaoBao would like to find the longest path from $(1, 1)$ to (n, n) such that each road is passed no more than once. Please help BaoBao find such a path.

Recall that if the sides of a triangle are of length a, b and c, we can infer that $a + b > c$, $a + c > b$ and $b + c > a$.

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains an integer n ($2 \le n \le 300$), indicating the size of the city.

For the following $(n - 1)$ lines, the i-th line contains i integers $a_{i,1}, a_{i,2}, \dots, a_{i,i}$ ($1 \le a_{i,j} \le 10^9$), where $a_{i,j}$ indicates the length of the road connecting intersection (i, j) and $(i + 1, j)$.

For the following $(n - 1)$ lines, the i-th line contains i integers $b_{i,1}, b_{i,2}, \dots, b_{i,i}$ ($1 \le b_{i,j} \le 10^9$), where $b_{i,j}$ indicates the length of the road connecting intersection (i, j) and $(i + 1, j + 1)$.

For the following $(n - 1)$ lines, the i-th line contains i integers $c_{i,1}, c_{i,2}, \dots, c_{i,i}$ ($1 \le c_{i,j} \le 10^9$), where $c_{i,j}$ indicates the length of the road connecting intersection $(i + 1, j)$ and $(i + 1, j + 1)$.

It's guaranteed that the sum of n of all test cases will not exceed 5×10^3.

Output

For each test case output three lines.

The first line contains one integer l, indicating the length of the longest path from $(1, 1)$ to (n, n) such that each road is passed no more than once.

The second line contains one integer m, indicating the number of intersections on the longest path.

The third line contains $2m$ integers $i_1, j_1, i_2, j_2, \dots, i_m, j_m$ separated by a space, where (i_k, j_k) indicates the k-th intersection on the longest path. Note that according to the description, there must be $(i_1, j_1) = (1, 1)$ and $(i_m, j_m) = (n, n)$.

If there are multiple valid answers, you can output any of them.

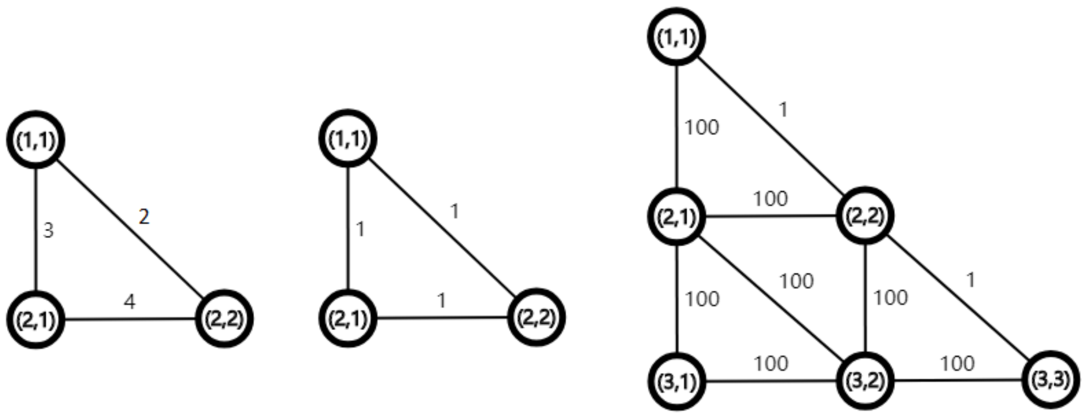
Please, DO NOT output extra spaces at the end of each line, or your solution may be considered incorrect!

Example

standard input	standard output
3 2 3 2 4 2 1 1 1 3 100 100 100 1 100 1 100 100 100	7 3 1 1 2 1 2 2 2 3 1 1 2 1 2 2 700 8 1 1 2 1 3 2 2 2 2 1 3 1 3 2 3 3

Note

The sample test cases are shown below:



Problem K. Happy Equation

Little Sub has just received an equation, which is shown below, as his birthday gift.

$$a^x \equiv x^a \pmod{2^p}$$

Given the value of a, please help Little Sub count the number of x ($1 \le x \le 2^p$) which satisfies the equation.

Input

There are multiple test cases. The first line of the input contains an integer T (about 1000), indicating the number of test cases. For each test case:

The first and only line contains two integers a and p ($1 \le a \le 10^9$, $1 \le p \le 30$).

Output

For each test case output one line containing one integer, indicating the answer.

Example

standard input	standard output
2 6 12 8 16	1023 16383

Problem L. Median

Recall the definition of the median of n elements where n is odd: sort these elements and the median is the $\frac{(n+1)}{2}$-th largest element.

In this problem, the exact value of each element is not given, but m relations between some pair of elements are given. The i-th relation can be described as (a_i, b_i), which indicates that the a_i-th element is strictly larger than the b_i-th element.

For all $1 \leq k \leq n$, is it possible to assign values to each element so that all the relations are satisfied and the k-th element is the median of the n elements?

Input

There are multiple test cases. The first line of the input contains an integer T, indicating the number of test cases. For each test case:

The first line contains two integers n and m ($1 \leq n < 100$, $1 \leq m \leq n^2$), indicating the number of elements and the number of relations. It's guaranteed that n is odd.

For the following m lines, the i-th line contains two integers a_i and b_i, indicating that the a_i-th element is strictly larger than the b_i-th element. It guaranteed that for all $1 \leq i < j \leq m$, $a_i \neq a_j$ or $b_i \neq b_j$.

It's guaranteed that the sum of n of all test cases will not exceed 2×10^3.

Output

For each test case output one line containing one string of length n. If it is possible to assign values to each element so that all the relations are satisfied and the i-th element is the median, the i-th character of the string should be '1', otherwise it should be '0'.

Example

standard input	standard output
2 5 4 1 2 3 2 2 4 2 5 3 2 1 1 2 3	01000 000

Note

For the first sample test case, as the 2nd element is smaller than the 1st and the 3rd elements and is larger than the 4th and the 5th elements, it's possible that the 2nd element is the median.

For the second sample test case, as the 1st element can't be larger than itself, it's impossible to assign values to the elements so that all the relations are satisfied.

Problem M. Sekiro

Sekiro: Shadows Die Twice is an action-adventure video game developed by FromSoftware and published by Activision. In the game, the players act as a Sengoku period shinobi known as Wolf as he attempts to take revenge on a samurai clan who attacked him and kidnapped his lord.

As a game directed by Hidetaka Miyazaki, Sekiro (unsurprisingly) features a very harsh death punishment. If the player dies when carrying g amount of money, the amount of money will be reduced to $\lceil \frac{g}{2} \rceil$, where $\lceil \frac{g}{2} \rceil$ indicates the smallest integer g' that $2g' \geq g$.

As a noobie of the game, BaoBao has died k times in the game continuously. Given that BaoBao carried n amount of money before his first death, and that BaoBao didn't collect or spend any money during these k deaths, what's the amount of money left after his k deaths?

Input

There are multiple test cases. The first line of the input contains an integer T (about 10^3), indicating the number of test cases. For each test case:

The first and only line contains two integers n and k ($0 \leq n \leq 10^9$, $1 \leq k \leq 10^9$), indicating the initial amount of money BaoBao carries and the number of times BaoBao dies in the game.

Output

For each test case output one line containing one integer, indicating the amount of money left after k deaths.

Example

standard input	standard output
4 10 1 7 1 10 2 7 2	5 4 3 2

Note

For the third sample test case, when BaoBao dies for the first time, the money he carries will be reduced from 10 to 5; When he dies for the second time, the money he carries will be reduced from 5 to 3.