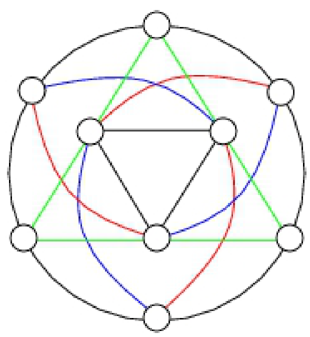
ANZAC 2015

Round 4

A Selection of Easy-to-Hard ACM Regional Problems

Compiled by Michael J. Dinneen (University of Auckland)



Notes:

1. For all problems, your programs must read the input from stdin/keyboard and output to stdout/console. No blank lines will appear in the input nor should appear in the expected output.
2. The problems, presented in no particular order of difficulty, have time limits of 5 seconds.
3. All of these problems have been sourced from different places on the Internet. We do not reveal their identity here, but none-the-less, credit, authorship, intellectual property, copyright, etc, remains with the original problem setters.

Problem A A Brick in Wall

After years as a brick-layer, you've been called upon to analyze the structural integrity of various brick walls built by the ACM Corporation. Instead of using regular-sized bricks, the ACM Corporation seems overly fond of bricks made out of strange shapes. The structural integrity of a wall can be approximated by the fewest number of bricks that could be removed to create a gap from the top to the bottom. Can you determine that number for various odd walls created?

Input

Input to this problem will begin with a line containing a single integer X ($1 \leq X \leq 100$) indicating the number of data sets. Each data set consists of two components:

- A single line, "MN" ($1 \leq M, N \leq 20$) where M and N indicate the height and width (in units), respectively, of a brick wall;
- A series of M lines, each N alphabetic characters in length. Each character will indicate to which brick that unit of the wall belongs to. Note that bricks will be contiguous; each unit of a brick will be adjacent (diagonals do not count as adjacent) to another unit of that brick. Multiple bricks may use the same characters for their representation, but any bricks that use identical characters will not be adjacent to each other. All letters will be uppercase.

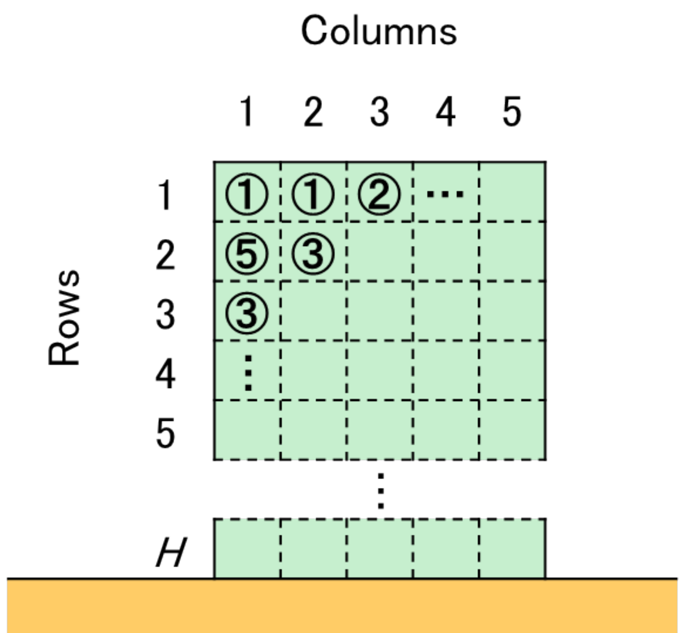
Output

For each data set, output the fewest number of bricks to remove to create a gap that leads from some point at the top of the wall, to some point at the bottom of the wall. Assume that bricks are in fixed locations and do not "fall" if bricks are removed from beneath them. A gap consists of contiguous units of removed bricks; each unit of a gap must be adjacent (diagonals do not count) to another unit of the gap.

Sample Input	Sample Output
3 5 7 AABBCCD EFFGGHH IIJJKKL MNNOOPP QQRRSST 5 7 AABBCCD AFFBGGD IIJBKKD MNNOOPD QQRRSST 6 7 ABCDEAB ABCFEAB AEAABAB ACDAEEB FFGAHIJ KLMANOP	5 2 2

Problem B Brocken Chain

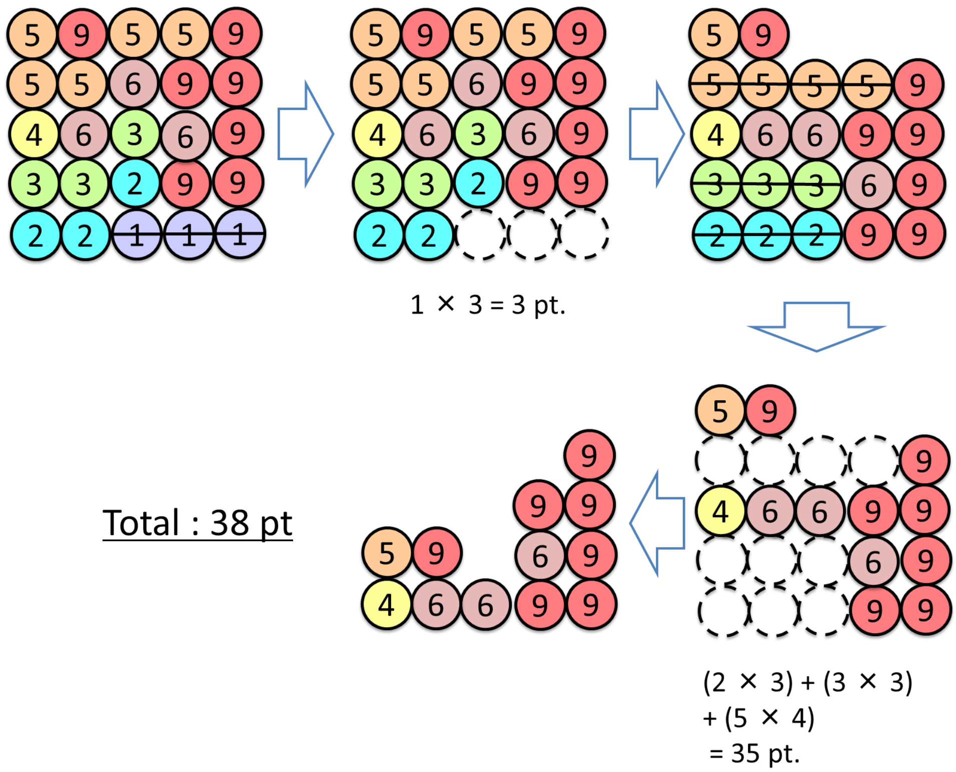
This chain puzzle, as shown in the figure below, H consists of various rows each with 5 columns of cells. Stone numbers from 1 to 9 are engraved and placed in each cell. If there is a stone carved with he same number in the adjacent cell in three or more horizontal row, these stones will disappear. The cells above extinguished stones fall down sequentially.



The puzzle proceeds according to the following steps.

1. If there is a stone carved with the same number in the adjacent cell in three or more horizontal, these stones will disappear. All of these annihilations of stone groups occur at the same time.
2. If over the cells where a stone is extinguished, stones fall to fill the empty spaces.
3. After dropping completion of all the stones, if a stone group satisfies the condition of disappearance, we repeat the process of step 1.

The score of this puzzle is the sum of the numbers of disappeared stones. We want a program to compute this score.



Input

The input consists of multiple data sets. Each data set has the following form

Height of the board H
Stone sequence of row 1
Stone sequence of row 2
...
Stone sequence of row H

The first line of each data set, the puzzle board height H is specified ($1 \leq H \leq 10$). A sequence of each row of stones has been specified from top to bottom. A sequence of stones is given by five numbers from 1 to 9, separated by a space. Numbers have been carved into the five stones of the line in that order.

The end of the input is indicated by the line consisting of a single zero (0).

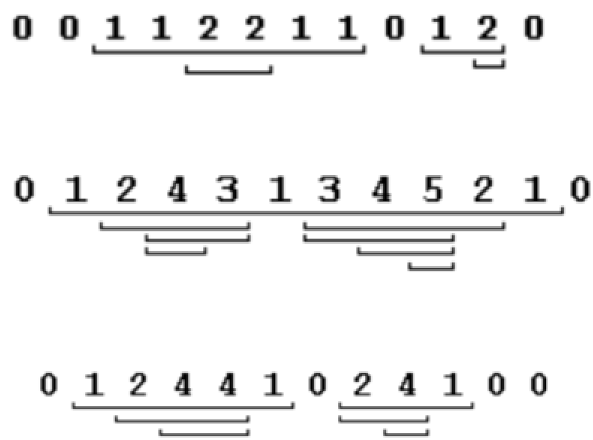
Output

Output a non-negative integer score for each puzzle case.

Sample Input	Sample Output
1 6 9 9 9 9 5 5 9 5 5 9 5 5 6 9 9 4 6 3 6 9 3 3 2 9 9 2 2 1 1 1 10 3 5 6 5 6 2 2 2 8 3 6 2 5 9 2 7 7 7 6 1 4 6 6 4 9 8 9 1 1 8 5 6 1 8 1 6 8 2 1 2 9 6 3 3 5 5 3 8 8 8 5 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 8 9 1 2 3 4 5 6 7 3 2 2 8 7 4 6 5 7 7 7 8 8 9 9 9 0	36 38 99 0 72

Problem C Current Islands

Given a sequence of integers $a_1, a_2, \ldots, a_n$, an *island* in a sequence is a contiguous subsequence for which each element is greater than the elements immediately before and after the subsequence. In the examples below, each island in the sequence has a bracket below it. The bracket for an island contained within another island is below the bracket of the containing island.



Write a program that takes as input a sequence of 12 non-negative integers and outputs the number of islands in the sequence.

Input

The first line of input contains a single integer P, ($1 \leq P \leq 1000$), which is the number of data sets that follow. Each data set, comprising of a sequence of space-separated 12 integers on one line, should be processed independently.

Output

For each data set there is one line of output. The single output line consists of the data set number K followed by a single space followed by the number of islands in the sequence.

Sample Input	Sample Output
4 1 0 0 1 1 2 2 1 1 0 1 2 0 2 0 1 2 4 3 1 3 4 5 2 1 0 3 0 1 2 4 4 1 0 2 4 1 0 0 4 0 1 2 3 4 5 6 7 8 9 10 0	1 4 2 8 3 6 4 10

Problem D Rout or Retreat

While the Romans may have had a fancy number system, the Visigoths (led by Alareiks, known now as Alaric I) managed to take Rome on August 24, 410. This was the first time that Rome had been taken by foreign troops in eight hundred years.

You will simulate a considerably less bloody (and less realistic) rout of soldiers by a potentially overwhelming force. Consider a group of invaders of strength I, and an opposing force of strength J, with distance D between them and a defensive stronghold of strength S. The Routing Force F of the invaders can be calculated as their strength times the distance:

$$F = ID$$

and the Blocking Force B of the defenders can be calculated as their strength times the square of their defenses:

$$B = JS^2$$

If F is less than or equal to B, the invaders' rout fails, and they are driven off. If F is greater than B, the rout is successful, all defenders of the stronghold are driven off, and the invaders can continue. However, their strength is reduced by removing a fraction equal to the ratio of the defenders' Blocking Force to their Routing Force for that attack. The resulting number should be rounded up to the nearest integer, to ensure that a successful rout leaves at least one soldier:

$$I_{new} = \lceil I(1 - B/F) \rceil$$

If the invaders make it through a stronghold, its location is considered the starting point for calculations regarding the next group of defenders, and so on, until either the invading force makes it all the way through the defenders or is completely routed.

Your goal is to determine whether or not a given invading force can make it through a particular gauntlet of defenders.

For the purposes of simplification, all defending strongholds in this simulation are considered to be in a straight line from the starting position of the invaders, and must be encountered in order from nearest to furthest away. No strongholds will be in the same location in a given set of data.

Input

Input to this problem will begin with a line containing a single integer N ($1 \leq N \leq 100$) indicating the number of data sets. Each data set consists of the following components:

- A line containing a single integer E ($1 \leq E \leq 20$) indicating the number of defensive strongholds in the data set;
- A series of E lines, each with three integers D, J, S ($1 \leq D, J \leq 10000$; $1 \leq S \leq 50$) separated by spaces representing the strongholds. D is the distance of the stronghold from the invaders' starting position; remember that all strongholds are considered to be in a straight line from the invaders' starting position, and must be encountered in order from nearest to furthest away. J and S represent the strength of the defenders and the stronghold, respectively, as in the above equation; and
- A line containing a single integer I ($1 \leq I \leq 30000$) representing the strength of the invading force.

Output

For each data set, print “ROUT!” if the invaders make it through all of the strongholds, or “RETREAT!” if the invaders cannot make it through all of the strongholds.

Sample Input	Sample Output
2 1 10 10 5 100 2 75 100 5 10 10 5 50	ROUT! RETREAT!

Problem E Bridge Removal

ICPC Islands was once a popular tourist destination. This time, for nature protection, we want to eliminate prohibited people of intrusiveness and of all artificial facilities. Most difficult in this project of is to remove all of the bridges that connects the islands.

The work team to dismantling the bridges of n islands is to start from any favorite island and can take any of the actions repeated below.

- Move across a bridge to an adjacent island.
- Remove a bridge to an adjacent island and stay on current island.

Note once a bridge has been removed, you are not be able to cross in either direction. When you cross a bridge or if you want to remove a bridge, it takes the time that is proportional to the length of the bridge.

We want to seek the shortest time required to remove all of the bridges between islands. The starting (favorite) and finishing island may be different.

Input

The input consists of up to 100 test cases. Each case has the following format.

n
$P_2 P_3 \ldots P_n$
$D_2 D_3 \ldots D_n$

Here n is the number of islands ($3 \leq n \leq 800$), labeled 1 to n. There are $n-1$ bridges connecting the islands; the connections are (i, P_i) where $1 \leq P_i \leq i-1$ for $i \geq 2$. The length of each bridge i is D_i.

The end of the input is indicated by a line containing only zero.

Output

For each data set, output the unit of time (shortest required) to remove all of the bridges, on a line by itself.

Sample Input	Sample Output
4 1 2 3 10 20 30 10 1 2 2 1 5 5 1 8 8 10 1 1 20 1 1 30 1 1 3 1 1 1 1 0	80 136 2

Problem F Number Guess

My tech-lover son has been attracted by the exciting game "Number Guess". In this game, the player must find a hidden positive integer number by at most T guesses (or turns). The parameter T together with a health parameter H is determined at the beginning of the game. In each turn, the player must enter a number. If the number is equal to the hidden number, he wins provided that $H \geq 0$. If the number is bigger than the hidden number, H is decreased by 1 unit of health. Otherwise, H remains unchanged. When H becomes negative or T reaches 0, the player definitely loses. The player can see the remaining turns and units of health after each turn. Although the game seems to be constructive, but something makes me suspicious! My son always wins. He claims he has a searching algorithm to find the hidden number but I can't believe him as for the given H and T there must be a big number which is not guessable by any searching algorithm. To prove my son claim is wrong, I kindly ask you to help me find the smallest M for which at least a number from 1 through M as the hidden number can't be guessed for the given T and H. For example, there is not any algorithm for finding all positive integers not greater than $M = 3$ by 2 turns and 0 units of health.

Input

There are multiple test cases. Each test case consists of one line containing two non-negative integers T and H ($0 \leq T, H \leq 100$). The input terminates with "0 0" which should not be processed.

Output

For each test case output M described above in one line. As M maybe too large, output it modulo $10^9 + 7$.

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

Problem G Summing Subsets

For a given set X of n not-necessarily-distinct numbers and a given number t, the goal is to compute the number of non-empty subsets Y of X with the properties that the sum over all members of Y is at most t and adding any member in $X \setminus Y$ to Y makes the summation greater than t. Note that the numbers in the set may have the same values, but they must be considered inherently different.

Input

There are multiple test cases in the input. Each test case starts with a line containing two non-negative integers $0 \leq n \leq 30$ and $0 \leq t \leq 1000$. The remainder of each test case consists of one or more lines containing n non-negative numbers belonging to X. The input terminates with "0 0" which should not be processed.

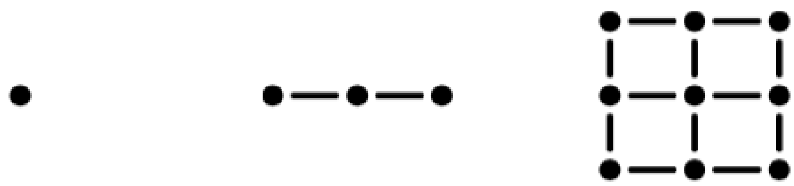
Output

For each test case, output the number of subsets defined above.

Sample Input	Sample Output
6 25 8 9 8 7 16 5 30 250 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 0 0	15 16509438

Problem H Spaceship

We need to solve a problem of maximizing the utilization of cargo spaceships. The ships transport valuable items that have the form of D-dimensional mesh with the size of 3 nodes in each dimension. The nodes are formed by balls of the same weight. The connections between balls are so light that their weight is negligible compared to the weight of balls. This means that the weight of any item is determined solely by the number of its nodes. On the other hand, the value of such an item is equal to the number of nodes plus the number of connections.



zero dimension weight = 1 price = 1 | first dimension weight = 3 price = 5 | second dimension weight = 9 price = 21

Each spaceships has a limited tonnage and we want to maximize the total value of items that may be stored in the ship without exceeding the tonnage. Your task is to decide what items of what dimension should be loaded to maximize their total value, providing that we have an unlimited supply of items of all dimensions.

Input

The first line of the input contains the number of test cases N. Each test case then consists of a single line containing one positive integer number $K < 10000000$ giving the ship cargo capacity.

Output

For each test case, print one line containing space-separated non-negative numbers $X_m X_{m1} \ldots X_1 X_0$, where $X_m > 0$ and X_i $(0 \leq i \leq m)$ is the number of items of the i-th dimension that we need to store to *maximize* their total value.

Sample Input	Sample Output
4	1
1	1 0 2 0 1
100	2 0 1 1 1
175	1 1 1 1 1 1 1 1 1
9841	

Problem I Wierd Numbers Revisited

In a previous programming contest, a problem called *Weird Numbers* dealing with numeric systems using a negative base was given. The problem assignment said:

A number N written in the system with a positive base R will always appear as a string of digits between 0 and $R-1$, inclusive. A digit at the position P (positions are counted from right to left and starting with zero) represents a value of R^P. This means the value of the digit is multiplied by R^P and values of all positions are summed together. For example, if we use the octal system (radix $R = 8$), a number written as 17024 has the following value:

$$1 \cdot 8^4 + 7 \cdot 8^3 + 0 \cdot 8^2 + 2 \cdot 8^1 + 4 \cdot 8^0 = 1 \cdot 4096 + 7 \cdot 512 + 2 \cdot 8 + 4 \cdot 1 = 7700$$

With a negative radix $-R$, the principle remains the same: each digit will have a value of $(-R)^P$. For example, a negaoctal (radix $-R = -8$) number 17024 counts as:

$$1 \cdot (-8)^4 + 7 \cdot (-8)^3 + 0 \cdot (-8)^2 + 2 \cdot (-8)^1 + 4 \cdot (-8)^0 = 1 \cdot 4096 - 7 \cdot 512 - 2 \cdot 8 + 4 \cdot 1 = 500$$

One big advantage of systems with a negative base is that we do not need a minus sign to express negative numbers. A couple of examples for the negabinary system ($R = -2$):

decimal	negabinary	decimal	negabinary	decimal	negabinary
-10	1010	-3	1101	4	100
-9	1011	-2	10	5	101
-8	1000	-1	11	6	11010
-7	1001	0	0	7	11011
-6	1110	1	1	8	11000
-5	1111	2	110	9	11001
-4	1100	3	111	10	11110

You may notice that the negabinary representation of any integer number is unique, if no "leading zeros" are allowed. The only number that can start with the digit "0" is the zero itself.

Today, we are interested whether there were any contestants' answers in that previous contest that were almost correct, i.e., their program output was different from the correct answer only by one. Will you help us to find out?

Input

The input contains several test cases. Each test case is given on a single line containing number X written in the negabinary notation. The line contains N ($1 \leq N \leq 10^6$) characters "0" or "1" representing the negabinary bits $a_{N1} \ldots a_1 a_0$ respectively. Numbers will be given to you without leading zeros, i.e., for each input where $X \neq 0$ it holds that $a_{N1} = 1$.

Output

For each test case, print a single line with number $(X + 1)$ written in the negabinary notation. Output the number without any leading zeros.

Sample Input	Sample Output
1 0 100 11 10101	110 1 101 0 1101010

Problem J Jive Racing

We have car races that are quite unusual. There are n cars participating in a race on the long straight track. Each car moves with a speed of 1 meter per second. Track has coordinates in meters. The car number i moves between two points on the track with coordinates a_i and b_i starting at the second 0 in the point a_i. The car moves from a_i to b_i, then from b_i to a_i, then from a_i to b_i again, and so on.

Handsome Mike wants to knock some cars out of the race using dynamite. Thus he has m questions. The question number j is: what is the number of cars in the coordinates between x_j and y_j inclusive after t_j seconds from the start?

Your task is to answer Mike's questions.

Input

The first line of the input contains two integers n and m ($1 \le n, m \le 1000$); the number of cars in the race and the number of questions.

Each of the following n lines contains a description of the car: two integers a_i and b_i ($0 \le a_i, b_i \le 10^9$, $a_i = b_i$); the coordinates of the two points between which the car i moves.

Each of the following m lines contains a description of the question: three integers x_j, y_j, and t_j ($0 \le x_j \le y_j \le 10^9$, $0 \le t_j \le 10^9$); the coordinate range and the time for the question j.

Output

Write m lines to the output file. Each line must contain one integer; the answer to the corresponding question in order they are given in the input file.

Sample Input	Sample Output
5 5	5
0 1	1
0 2	2
2 3	4
3 5	3
4 5	
0 5 0	
0 1 2	
0 2 1	
2 5 2	
2 5 3	