Problem A. K-based numbers

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Let's consider K-based numbers, containing exactly N digits. We define a number to be valid if its K-based notation doesn't contain two successive zeros. For example:

- 1010230 is a valid 7-digit number;
- 1000198 is not a valid number;
- 0001235 is not a 7-digit number, it is a 4-digit number.

Given two numbers N and K, you are to calculate an amount of valid K based numbers, containing N digits.

Input

The numbers N and K in decimal notation separated by the line break. ($2 \le K \le 10$, $2 \le N$, $N+K \le 18$)

Output

Output one integer — the result in decimal notation.

Examples

standard input	standard output
2 10	90

Problem B. Stones

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

N stones lie on a table. There are tow players. On each move, player can take:

- 1 or 2 stones if N is divided by 3;
- 1 or 3 stones if $N - 1$ is divided by 3;
- 1, 2 or 3 stones if $N - 2$ is divided by 3;

Each move can be done only if there are enough stones to take. The player who cannot move, lose the game.

The question is: which player has an optimal strategy to win?

Input

One integer N ($1 \leq N \leq 100$).

Output

Print 1 or 2 — number of player winning the game

Example

standard input	standard output
1	1

Problem C. Roads to cinematography

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 mebibytes

After the fiasco in Santa Carolina Mr. John First and his friends opened a new shrine of cinematograpy. Their new promotion location was chosen not far from a major railroad. Black Jack offered to build roads to the nearest stations where passenger trains stop.

Mr. John First drew a coordinate system. The origin point was their current location, and the axes X and Y were oriented to the east and north, respectively. Stations were marked as points with the coordinates (x_i, y_i), where $i = 1, 2, \ldots, n$. The railway passes in the southeast direction, so the following inequations hold true:

$$\begin{cases} 0 \le x_1 \le x_2 \le x_3 \le \ldots \le x_{n-1} \le x_n \\ 0 \le y_n \le y_{n-1} \le y_{n-2} \le \ldots \le y_2 \le y_1 \end{cases}$$

Diana insists that each road should be oriented strictly at one of the cardinal directions, i.e. that the corresponding segment should be oriented to one of the coordinate axes. Jack says that roads must be built so that each station is connected by cinema directly or indirectly with those roads (without using the railroad) and that the distance between each station and the cinema is minimal. The distance must be minimal provided that the roads are parallel to the coordinate axes.

Roads cost money, and Mr. John First wants the total length of the roads to be minimal, provided that Diana's and Jack's conditions are met. The money they'd save could be spent on new films.

Input

The first line of the input file contains an integer n — the number of stations ($1 \le n \le 500$). Each of the following n lines contains two integers x_i, y_i — the coordinates of a station ($0 \le x_i, y_i \le 10^6$).

It is guaranteed that the sequence (x_i) is nonstrictly ascending and the sequence (y_i) is nonstrictly descending. No two stations are located in the same point. No station is located in the origin point.

Output

In the first line of the output file, print two integers: k — the number of roads to be built ($1 \le k \le 2000$) and A — the total length of these roads. Each of the following k lines must describe a road.

A road description must contain four integers x_1, y_1, x_2, y_2 — the coordinates of the end points of the road ($x_1 \ge x_2$, $y_1 \ge y_2$). One of the following two statements must hold true: either $x_1 = x_2$, or $y_1 = y_2$.

It is guaranteed that an optimal layout exists which meets the limitations of the output data format. If several solutions are possible, print any of them.

Example

standard input	standard output
3 0 9 2 4 5 1	4 16 0 9 0 4 2 4 0 4 5 1 0 1 0 4 0 0

Problem D. Martian Army

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

Many centuries ago Martians switched to using huge robots for military operations. During the current Moon conquest campaign, all of the Martian army is located at the headquarters on Mars, and each person is controlling the actions of his robot. There is a strict hierarchy in the Martian army: each person excepting the general (there is only one general in the army) has a direct commander. According to the army regulations, communication is allowed only between a commander and his direct subordinate. The communication is carried out via the headquarters local network. At the headquarters, each military person has his own computer, and computers are numbered from 1 to N, where N is the size of the Martian army. It is a tradition that a subordinate's computer has number that is greater than the number of his commander's computer. Each military person, in addition to the number of his computer, is characterized by his reliability. This is a real number; the owner of the computer i has reliability A_i. The general has reliability 1, and soldiers (soldiers and only they have no subordinates) have reliability 0.

The traffic in the headquarters network is not free for the military. For every megabyte of traffic between the i-th computer and the computer of the commander of the i-th computer's owner, the central Martian provider demands C_i Martian dollars in payment. The complication is that the volume of traffic between any two headquarters computers is a state secret, and is unknown even to the provider. Every month the provider sends a bill, and the military write there the traffic (a whole number of megabytes) themselves. Let a commander and his subordinate have computers with numbers i and k respectively. According to the contract with the provider, the traffic between the computers i and k must be not less than A_i˘A_k. In the beginning of every month, the provider's representatives know the hierarchy in the Martian army and costs of a megabyte of traffic, but they don't know the numbers A_i except for the general and soldiers, and of course they don't know beforehand the amounts of traffic that the military will write in the bill. It is interesting to know the guaranteed amount of money that the provider will receive from the military.

Input

The first line contains the size of the army $2 \leq N \leq 100000$. Each of the next N˘1 lines contains integers K_i and C_i, which are the number of the computer of the commander of the i-th computer's owner and the cost of a megabyte of traffic between the computers i and K_i ($1 \leq K_i < i \leq N, 0 \leq Ci \leq 1000$).

Output

Output the minimal guaranteed amount of payment to the provider in Martian dollars. This amount must be a real number given with exactly two decimal digits.

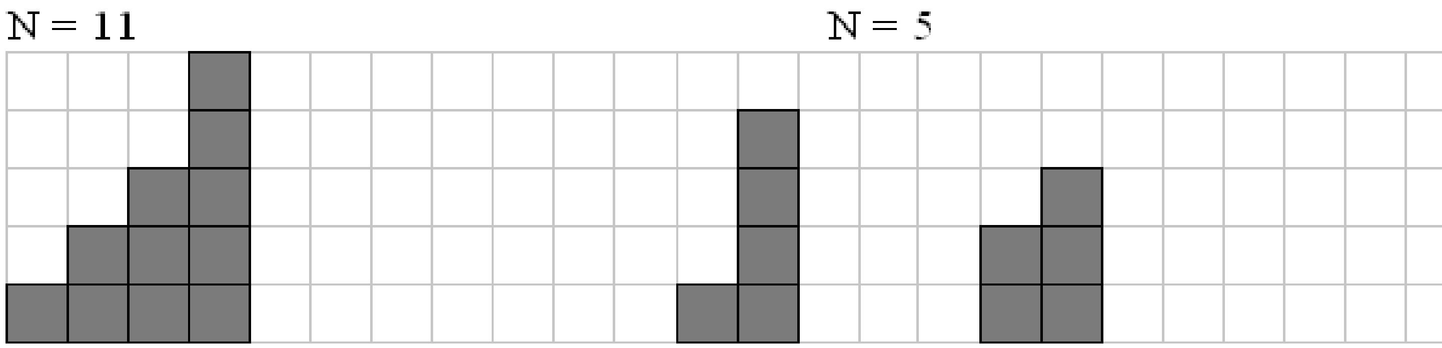
Examples

standard input	standard output
7 1 10 2 5 2 3 3 1 3 2 3 3	8.00

Problem E. Staircases

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

One curious child has a set of N little bricks. From these bricks he builds different staircases. Staircase consists of steps of different sizes in a strictly descending order. It is not allowed for staircase to have steps equal sizes. Every staircase consists of at least one step and each step contains at least one brick. Picture gives examples of staircase for N=11 and N=5:



Your task is to write a program that reads the number N and writes the only number Q — amount of different staircases that can be built from exactly N bricks.

Input

The only integer N $(1 \leq N \leq 100)$ — the amount of bricks.

Output

Print one integer Q — the answer to the problem.

Examples

standard input	standard output
1	1
7	5
100	444793

Problem F. Vertex permutation

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	64 mebibytes

You are given an undirected graph of n vertices and m edges. You assign number from 1 to n to each vertex in a such way that no two vertices have equal numbers assigned to them. Your task is to minimize a function:

$$S = \sum_{(u,v) \in E} abs(p_u - p_v),$$

where E is the set of all edges, p_u and p_v are numbers assigned to vertices u and v.

Input

The first line of input contains two integers n and m ($1 \le n \le 20$, $1 \le m \le 100$). Each of the next m lines contains two integers u_i and v_i and describes edge of the graph. All edges are different and $1 \le u_i < v_i \le n$.

Output

Output the minimum possible value of S.

Examples

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

Problem G. Pizza Delivery Guy

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	256 mebibytes

You deliver pizzas for living. You have a knapsack of size S and very big order of n pizzas, i-th one has size a_i. Your task is it to delivery pizzas as soon as possible. But you have neither vehicle, nor friends to help you, the only way is to distribute all the pizzas into some piles of summary size no more S each and deliver the piles by hands one by one. So you have to minimize number of piles of pizzas.

Input

Input consists of t tests ($1 \le t \le 10$). The first line of the file contains integer t. Rest of the input contains t tests. Each test is described by two lines: integers n ($1 \le n \le 20$) and S ($1 \le S \le 10^9$) on the first line, and integers $a_1, a_2, \ldots, a_n$ on the second one. All a_i are from 1 to S.

Output

For each test output minimal number of piles m and then m lines, which describe piles. i-th pile should be described as number of pizzas k_i and indexes of the pizzas. Each pizza should be at exactly one pile. If there are several optimal solutions, output any of them.

Examples

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

Problem H. Jetpack

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	64 mebibytes

Little Mirko got a new mobile phone for his birthday! As all kids nowadays, he quickly downloaded all of the popular mobile games, including Jetpack Joyride. In the game, the protagonist Barry is running across a field consisting of 10 rows and N columns of squares of equal size. Initially, Barry is located in the center of the square in the lower left corner. Barry is constantly running to the right at the speed of one square per second. Additionally, he must avoid obstacles that are in his way.

When Mirko presses the phone screen, Barry turns on his super-duper special jetpack and starts his ascent at the speed of one square per second (still moving to the right, now moving diagonally up at an angle of 45°, until he reaches the ceiling, when he will continue moving to the right until Mirko releases the screen). When Mirko releases the phone screen, Barry starts falling down at the speed of one square per second (now moving diagonally again, but this time facing down, until he reaches the floor, when he will continue moving to the right).

Mirko just started playing the game recently and he's still not good at it. He saw on YouTube that someone managed to complete the game by crossing all N columns, so he is asking you for your help. He will give you the layout of the fields in the game, and you must output the moves he has to play in order to win.

Input

The first line of input contains the integer N ($1 \le N \le 10^5$), the size of the field. Each of the following 10 lines contains N characters '.' and 'X', the layout of the field in the game. The characters 'X' denote obstacles, and '.' walkable fields.

Output

The first line of output must contain the integer P ($0 \le P \le 5 \cdot 10^4$), the number of moves Mirko has to make. In the following P lines, output any series of P moves, each in its own line, such that it solves Mirko's problem from the task.

A move is determined by two integers t_i and x_i, where t_i denotes the second in which Mirko has to press the screen, and x_i denotes how long he needs to keep the screen pressed.

A series of moves must be sorted in chronological order. In other words, it must hold $t_i + x_i \le t_{i+1}$. Also, no move should begin after the end of the game, $t_i < N$. The input data will be such that a solution will surely exist.

Example

standard input	standard output
11 .....XX...X ....XX...XX ...XX...XX. ....XXX.... ....X..... ....XX...X. ...XX...XX. ...X...XX..	2 1 4 7 2

Note

Clarification of the first test case: The path Mirko has to take is denoted with '*':

```
.....XX...X
....XX...XX
...XX...XX.
...........
....XXX....
.....*...*.
....*X*.*.*
...*XX.*.X.
..*XX...XX.
**.X...XX..
```

Problem I. Number Of Combination With Repeats

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Combination with repeats is some way to choose k elements out of n, so that ways that differ only by permutation of its elements are considered equal. There are more combinations with repeats than normal combinations because you can choose some element more than once. This type of combinations can be represented as a vector of length k with all elements from 1 to n. The elements are in non-decreasing order. For example, there are 6 combinations with repeats with k elements taken out of n elements.

In this problem we consider all k-element combinations with repeats of set of n numbers from 1 to n. Of course, we will sort them all lexicographically as vectors (we assume that elements in combination are non-decreasing). For example, for $n = 3$ and $k = 2$ the sorted list of all combinations with repeats looks like this: $(1, 1)$, $(1, 2)$, $(1, 3)$, $(2, 2)$, $(2, 3)$, $(3, 3)$.

In this problem you are required to find the number of given combination with repeats (enumeration start with 0).

Input

First line contains two integer n and k ($1 \leq k \leq n \leq 30$). Second line contains k integers in non-decreasing order, representing a combination with repeats.

Output

Output the number of the given combination with repeats in lexicographical order (starting with zero).

Examples

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

Problem J. Not So Deep Sequences

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	256 mebibytes

Polina wrote a program deleting all the symbols except «(» and «)» from the given string. Now she's interested in how many different correct bracket sequences of length $2n$ she can get.

It is known, that Polina only runs her program on correct mathematical expressions, where the maximum number of nested parentheses is exactly k.

Input

The only line of the input contains two integers n ($1 \le n \le 50$) and k ($1 \le k \le n$).

Output

Print the only number — number of correct bracket sequences modulo $10^9 + 7$.

Examples

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