NTU IEEE Code Jam 2013

Website: http://acm.hust.edu.cn/vjudge/contest/toListContest.action
Search for Contest Title: IEEE Code Jam 2013
Contest password: IEEE 2013

Languages permitted: C, C++, Python, Java

A - K-based Numbers

Time Limit:1000MS **Memory Limit:**16384KB **64bit IO Format:**%I64d & %I64u

Description

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.
You may assume that $2 \le K \le 10$; $N \ge 2$; $N + K \le 18$.

Input

The numbers *N* and *K* in decimal notation separated by the line break.

Output

The result in decimal notation.

Sample Input

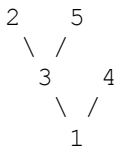
input	output
2 10	90

B - Binary Apple Tree

Time Limit:1000MS **Memory Limit:**65536KB **64bit IO Format:**%I64d & %I64u

Description

Let's imagine how apple tree looks in binary computer world. You're right, it looks just like a binary tree, i.e. any biparous branch splits up to exactly two new branches. We will enumerate by integers the root of binary apple tree, points of branching and the ends of twigs. This way we may distinguish different branches by their ending points. We will assume that root of tree always is numbered by 1 and all numbers used for enumerating are numbered in range from 1 to N, where N is the total number of all enumerated points. For instance in the picture below N is equal to 5. Here is an example of an enumerated tree with four branches:



As you may know it's not convenient to pick an apples from a tree when there are too much of branches. That's why some of them should be removed from a tree. But you are interested in removing branches in the way of minimal loss of apples. So your are given amounts of apples on a branches and amount of branches that should be preserved. Your task is to determine how many apples can remain on a tree after removing of excessive branches.

Input

First line of input contains two numbers: N and Q ($2 \le N \le 100$; $1 \le Q \le N - 1$). N denotes the number of enumerated points in a tree. Q denotes amount of branches that should be preserved. Next $N - 1$ lines contains descriptions of branches. Each description consists of a three integer numbers divided by spaces. The first two of them define branch by it's ending points. The third number defines the number of apples on this branch. You may assume that no branch contains more than 30000 apples.

Output

Output should contain the only number — amount of apples that can be preserved. And don't forget to preserve tree's root ;-)

Sample Input

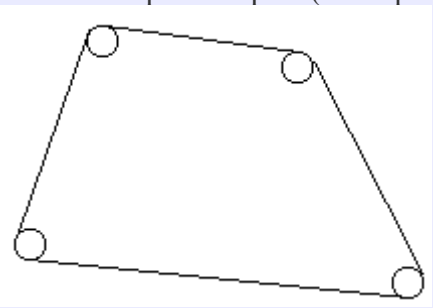
input	output
5 2 1 3 1 1 4 10 2 3 20 3 5 20	21

C - Rope

Time Limit:1000MS **Memory Limit:**16384KB **64bit IO Format:**%I64d & %I64u

Description

Plotters have barbarously hammered N nails into an innocent plane shape, so that one can see now only heads. Moreover, pursuing their mean object, they have hammered all the nails into the vertices of a convex polygon. After that they…it is awful… have roped off the nails, so that the shape felt upset (the rope was very thin). They've done it as it is shown in the figure.



Your task is to find out a length of the rope.

Input

There two numbers in the first line of the standard input: N — a number of nails ($1 \le N \le 100$), and a real number R — a radius of heads of nails. All the heads have the same radius. Further there are N lines, each of them contains a pair of real coordinates (separated by a space) of centers of nails. An absolute value of the coordinates doesn't exceed 100. The nails are described either in a clockwise or in a counter-clockwise order starting from an arbitrary nail. Heads of different nails don't adjoin.

Output

Output a real number with two digits precision (after a decimal point) — a length of the rope.

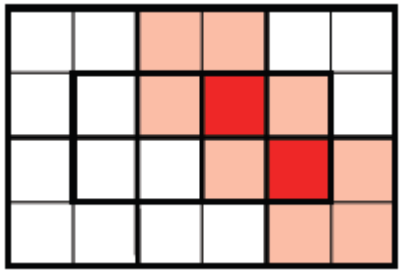
Sample Input

input	output
4 1 0.0 0.0 2.0 0.0 2.0 2.0 0.0 2.0	14.28

D - Glass Pyramid

Time Limit:1000MS **Memory Limit:**65536KB **64bit IO Format:**%I64d & %I64u

Description



Vova is a skyscraper fan. When he arrives in a city that has any skyscrapers, he always tries to get to the observation deck in the highest skyscraper to make several shots from there. This time is no exception. As soon as Vova arrived in Hong Kong, right after checking in a hotel, he got to the International Commercial Center, a 118-floor building that was built a couple of years ago. Before the main entrance into the International Commercial Center there is a pyramid of large glass blocks. Each block has the shape of a one meter height rectangular parallelepiped with a 2×2 meter square base. The pyramid's lowest layer is $m \times n$ meter large. The second layer is $(m - 2) \times (n - 2)$ meter large and is located so that the vertices of each block's lower side coincided with the centers of the upper sides of the previous layer. The third and the next layers are constructed similarly. The upper layer consists of a single row of blocks (specifically, at $m = n$ it consists of one block). Some pyramid blocks are made of colorless glass and the other blocks are made of reddish glass.

When Vova got to the observation platform, located at the skyscraper's 100-th floor, he saw the pyramid from there as a rectangular field, divided into 1×1 meter cells. The reddish hue of each cell hinted the number of reddish glass blocks that laid on this cell. Vova took the photo of the pyramid from above and now he wants you to take a look at this photo and find out which pyramid blocks are made of reddish glass and which ones are made of colorless glass. Vova also says that there is no reddish block lying exactly above a colorless block.

Input

The first line contains integers m and n that are sizes of the pyramid's base ($2 \le m, n \le 40$; m and n are even). Then m lines follow, each of them contains n non-negative integers, describing the photo Vova took. Each integer shows the number of reddish blocks lying above the corresponding cell. It is guaranteed that the input gives some possible arrangement of blocks in the pyramid.

Output

Print the description of the layers, from the lowest one to the highest one. To print information about the i-th layer, use k_i lines each containing l_i characters ($k_1 = m/2$; $l_1 = n/2$; $k_i = k_{i-1} - 1$; $l_i = l_{i-1} - 1$). Character "R" represents a reddish block, "W" represents a colorless block. Each next layer is separated from the previous one by an empty line. If there are multiple solutions, you can print any of them.

Sample Input

input	output
4 6 0 0 1 1 0 0 0 0 1 2 1 0 0 0 0 1 2 1 0 0 0 0 1 1	WRW WWR WR

E - Steaks on Board

Time Limit:1000MS **Memory Limit:**65536KB **64bit IO Format:**%I64d & %I64u

Description

The Oceanic Airlines company has announced that starting from the new year all passengers of transoceanic flights will be able to taste the famous Ural steaks during the flight. Unlike the standard meals, the steaks will be fresh, hot, and incredibly delicious because they will be cooked right on board.

Each passenger can specify at check-in the exact time when she wants her steak to be served. A steak is cooked by frying each of its sides on a frying pan continuously for one minute. If a steak is cooked too early, it won't be delicious enough, so the cooking must start no earlier than x minutes before the steak is served.

Unfortunately, each Oceanic Airlines plane is equipped with only one electric stove and one frying pan. No more than k steaks can be cooked on the frying pan simultaneously. The stove can be switched off when nothing is cooked. The chef wants to serve the orders of all the passengers on time while minimizing the time during which the stove is switched on. Help the chef.

Input

The first line contains the integers x and k ($2 \le x \le 1\,000$; $1 \le k \le 50$). In the second line you are given the number n of ordered steaks ($1 \le n \le 50$). In the third line there are integers t_1, …, t_n, which define the times when the steaks should be served ($2 \le t_i \le 1\,000$). The times are given in nondecreasing order and are counted from the moment of departure.

Output

If the chef can cook the ordered steaks on time, output in the first line the minimum number of minutes during which the electric stove will be working. In the ith of the following n lines output two nonnegative integers, which are the times when the ith steak should be put on the frying pan on one side and on the other side, respectively. The steaks must be described in the same order in which they are given in the input. If there are several possible answers, output any of them. If it is impossible to serve all the orders on time, output "-1".

Sample Input

input	output
10 2 3 2 16 25	5 0 1 11 15 15 17
10 2 4 7 8 9 10	4 3 6 4 5 3 6 4 5
2 1 2 2 2	-1

F - Parencodings

Time Limit:2000MS **Memory Limit:**65536KB **64bit IO Format:**%lld & %llu

Description

Let S = s1 s2 ... s2n be a well-formed string of parentheses. S can be encoded in two different ways:

- By an integer sequence P = p1 p2 ... pn where pi is the number of left parentheses before the ith right parenthesis in S (P-sequence).
- By an integer sequence W = w1 w2 ... wn where for each right parenthesis, say a in S, we associate an integer which is the number of right parentheses counting from the matched left parenthesis of a up to a. (W-sequence).

Following is an example of the above encodings:

S ((((()()())))
P-sequence 4 5 6666
W-sequence 1 1 1456

Write a program to convert P-sequence of a well-formed string to the W-sequence of the same string.

Input

The first line of the input contains a single integer t (1 <= t <= 10), the number of test cases, followed by the input data for each test case. The first line of each test case is an integer n (1 <= n <= 20), and the second line is the P-sequence of a well-formed string. It contains n positive integers, separated with blanks, representing the P-sequence.

Output

The output consists of exactly t lines corresponding to test cases. For each test case, the output line should contain n integers describing the W-sequence of the string corresponding to its given P-sequence.

Sample Input

```
2
6
4 5 6 6 6 6
9
4 6 6 6 6 8 9 9 9
```

Sample Output

```
1 1 1 4 5 6
1 1 2 4 5 1 1 3 9
```

G - Balloon

Time Limit:3000MS **Memory Limit:**65536KB **64bit IO Format:**%lld & %llu

Description

The weather is wonderful today. *Gao* takes a walk in the garden with his girlfriend. His girlfriend likes balloons so much, so that she wants to fly some balloons (not kites!) in the garden.

We can regard the garden as a three-dimensional space, and its coordinate starts from (0,0,0) to (10000,10000,10000). There are *N*groups of balloons, each of groups has a red balloon and a blue balloon. We can regard each balloon as a sphere, all the radius of spheres are *R*. The center of each sphere will be given in the input.

For some reasons, she wants to choose one balloon (red one or blue one) from **each** group, so that she can put exactly *N* balloons in the garden. It's obvious that there is no overlap for any two balloons in the *N* balloons which she selected. The largest *R* will make *Gao*'s girlfriend happiest. Can you help *Gao* to calculate the largest R?

Input

There are multiple cases. For each case, The first line contains an integer N ($2 \le N \le 200$), meaning there are *N* groups of balloons. In the next *N* lines, each line contains six integers, indicating the coordinates of two balloons.

NOTICE: The garden only limits the center of the balloon.

Output

For each test case, it contains one real number indicating the largest *R*. The results should be rounded to three decimal places. You should promise that there is still no overlap for any two balloons after rounded.

Sample Input

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

Sample Output

```
3.464
```

H - Box

Time Limit:2000MS **Memory Limit:**65536KB **64bit IO Format:**%lld & %llu

Description

Ivan works at a factory that produces heavy machinery. He has a simple job --- he knocks up wooden boxes of different sizes to pack machinery for delivery to the customers. Each box is a rectangular parallelepiped. Ivan uses six rectangular wooden pallets to make a box. Each pallet is used for one side of the box.

Joe delivers pallets for Ivan. Joe is not very smart and often makes mistakes --- he brings Ivan pallets that do not fit together to make a box. But Joe does not trust Ivan. It always takes a lot of time to explain Joe that he has made a mistake.

Fortunately, Joe adores everything related to computers and sincerely believes that computers never make mistakes. Ivan has decided to use this for his own advantage. Ivan asks you to write a program that given sizes of six rectangular pallets tells whether it is possible to make a box out of them.

Input

There are several test cases in the input. Each case consists of six lines. Each line describes one pallet and contains two integer numbers w and h ($1 \le w, h \le 10\,000$) --- width and height of the pallet in millimeters respectively.

Output

Write a single word "POSSIBLE" to the output file if it is possible to make a box using six given pallets for its sides. Write a single word "IMPOSSIBLE" if it is not possible to do so.

Sample Input

```
1345 2584
2584 683
2584 1345
683 1345
683 1345
2584 683
1234 4567
1234 4567
4567 4321
4322 4567
4321 1234
4321 1234
```

Sample Output

```
POSSIBLE
IMPOSSIBLE
```