A. Dance Floor

Manao owns a dance floor which hosts dances every Saturday night. A feature of his dance floor is that they also rent dance shoes. There are shoes of n different colors for rent. We will enumerate the colors from 1 to n. For each i, there are M_i pairs of men's shoes, W_i pairs of women's shoes and S_i pairs of shoelaces of color i.

Of course, each dancer wants to have the shoes and shoelaces of the same color. Determine the maximum number of pairs that can dance if each pair consists of a man and a woman. Note that the two dancers in the pair do not have to wear the shoes of the same color.

Input

The first line contains an integer n ($1 \le n \le 100000$). The i-th of the following n lines contains three space-separated integers M_i, W_i, S_i ($0 \le M_i, W_i, S_i \le 20000$).

Output

Print a single number in a single line — the maximum number of pairs of dancers that can dance in rented shoes.

Samples

1 5 7 5	2
1 5 7 12	5

In the first example, there are 5 pairs of men's shoes, 7 pairs of women's shoes and only 5 pairs of shoelaces, all of the same color. There can be no more than 2 pairs of dancers on the floor, because there are not enough shoelaces for another pair.

In the second example, there are now 12 pairs of shoelaces, which would be enough for 6 pairs. However, there are not enough men's shoes to form these pairs.

B. Palindrome

Manao and his friend like to ask each other brainteasers every other day. Recently, Manao's friend brought him a string s and claimed this string has been obtained from a palindrome (see notes for definition of palindrome) by picking two positions and exchanging characters at those positions. The friend asked Manao to determine the palindrome that the string was obtained from.

Manao suspects his friend could have cheated and the string could not be a result of the abovementioned process. Also, he noticed that there might be several possible answers to the problem posed. Help him out by finding all the possible answers, or determining that there is none.

Input

The single line contains the string s consisting of at least 2 characters. The string will contain lowercase letters only. The length of the string will not exceed 100000.

Output

In the first line, print a single number p - the number of different palindromes from which the given string can be obtained by exchanging letters at exactly two positions. Print these

palindromes in the next p lines, one answer per line. The palindromes should be ordered lexicographically.

Samples

abab	2 abba baab
abracacarba	1 abracacarba

In the first example, there are exactly two palindromes from which the string "abab" can be generated: "abba" and "baab".

In the second example, the only possible answer is the given string itself. Manao's friend could swap any two equal characters in it.

A palindrome is a word that reads the same forward and backward.

C. Hopscotch Race

Hopscotch races are played on a square grid of numbers of size $2N + 1$. The goal is to get to the center space in the fewest number of moves. A move consists of moving to an edge adjacent space where the signed difference between the new and current spaces is greater than or equal to the signed difference between the current and previous spaces. Players start at any space on the edge of the square with a previous difference of zero. Help Manao win the race by telling him the space to start in to reach the center in the fewest moves and the number of moves required. If multiple spaces work, return the lexicographically smallest. If there is no path, print “IMPOSSIBLE”.

Input

The first line contains a single integer N ($1 \le N \le 300$). $2N + 1$ lines follow. Each of them contains $2N + 1$ positive integers describing the board. All spaces will have values less than or equal to 100000000.

Output

If it is possible to reach the center of the grid from some edge then output two lines. The first of these lines should contain a single integer which is the minimal number of moves required. The second line should contain the row and column (both 1-based) of the lexicographically smallest starting location. If it is impossible, print “IMPOSSIBLE” in a single line.

Samples

1 1 2 3 4 5 6 7 8 9	1 1 2
1 3 3 3 3 2 3 3 1 3	1 3 2
2 9 9 1 9 9 9 9 4 9 9 9 9 6 9 9 9 2 4 9 9 9 2 1 9 9	3 5 2
2 3 1 2 4 5 1 3 4 5 4 2 4 5 4 2 4 5 4 3 1 5 4 2 1 3	IMPOSSIBLE

The spaces are numbered from 1 to $2N + 1$ starting in the top left corner. The "lexicographically smallest" space is the one with the smallest row. If multiple spaces work along the same row, it is the one with the smallest column.

D. Board Queries

Manao has a square board consisting of n rows and n columns. Each cell of this board can either contain a period ('.') or an 'X' character. Initially, the board is filled with periods.

Manao has to apply m queries to this board. The queries can be of three types:

- Rotate the entire board clockwise or counterclockwise by 90 degrees;
- Mirror the board horizontally or vertically;
- Replace some fragment of our board with a given 3x3 board.

Determine how the board will look like after all of the queries are applied.

Input

The first line contains two space-separated integers n and m ($3 \le n \le 1000$, $0 \le m \le 50000$). The following lines describe the queries in order of their application. The first line describing a query contains a single string 'ROTATE', 'MIRROR' or 'REPLACE'.

'ROTATE' will be followed by a single line that will contain either 'CW' or 'CCW'. The former means that the board should be rotated clockwise, and the latter means that board should be rotated counterclockwise.

'MIRROR' will be followed by a single line that will contain either 'HOR' or 'VER'. 'HOR' means that the board should be mirrored horizontally, i.e., against its middle row. As a result of this process, the first and the last row should get switched, as well as the second and the second-to-last, and so on. 'VER' means that the board should be mirrored vertically in the same fashion.

'REPLACE' will be followed by a line with two numbers i and j ($1 \le i, j \le n - 2$) and then by three lines containing three characters '.' or 'X' each. As a result of this operation, the subboard of our board with top left corner in (i, j) should get replaced with the board given in the three lines. i denotes the 1-based index of the row and j denotes the 1-based index of the column.

Output

Print the board resulting from Manao's queries.

Sample

4 3 REPLACE 2 2 XXX ..X XX. ROTATE CW MIRROR VER	.... .X.X .X.X .XX.

In the example, the board is initially empty. After the first operation, the board will look as follows:

```
....
.xxx
...x
.xx.
```

After that, the board is rotated clockwise:

```
....
x.x.
x.x.
.xx.
```

As a last step, the board is mirrored vertically.

E. Homework

Everyone needs help with his homework sometimes. Today is a rare occasion when the person in need is Manao.

Manao's homework looks as follows. He is given an array of distinct integers $a_1, a_2, ..., a_n$. He should pick exactly k distinct elements from it and increment each of them by 1. The goal is to maximize the following sum:

$$\sum_{i=1}^{n}\sum_{j=i}^{n}(j-i+1)\cdot min(a_i, a_{i+1}, ...a_j)$$

Help Manao by finding the maximum possible value of the sum that can be achieved under given constraints.

Input

The first line contains two space-separated integers n and k ($2 \le n \le 100000$, $0 \le k \le n$). The next line contains n distinct space-separated integers $a_1, a_2, ..., a_n$ ($0 \le a_i \le 100000$).

Output

Print a single number in a single line – the maximum possible sum Manao can achieve. Note that this value is guaranteed to fit within 64-bit signed integer under given constraints.

Samples

2 0 4 1	7
4 2 5 1 2 3	47

In the first example, the sum is equal to $a_1 + 2 \cdot min(a_1, a_2) + a_2$. Since Manao is not supposed to increment anything, the answer is $4 + 2 \cdot min(4, 1) + 1 = 7$.

In the second example, Manao can increase the second and third element of the array to obtain (5, 2, 3, 3), which has the maximum sum of all possible arrays he can obtain.

F. Railroad Tycoon

Manao's uncle recently passed away and left Manao his railroad empire. There are k distinct tracks connecting two cities. Each track has n_i refueling stations. If any refueling station on a track is not functioning, it is impossible for a train to pass between the cities on that track. Unfortunately, Manao is terrible at managing the railroad empire and the employees threaten to go on strike. They tell Manao that at the end of

each day the employees at a random open refueling station will close it and leave it closed till the end of the strike. They say this will continue until their demands are met or all refueling stations are closed. Manao does not care about the railroad business anyways, but he does want to make some quick cash. To effectively sell tickets he needs to know the expected number of days that there will remain a functioning track between the two cities.

Input

The first line will contain one integer k the number of tracks ($1 \le k \le 20$ and $1 \le n_i \le 100000000$). The second line will contain k integers denoting the number of refueling station on each track.

Output

A single floating point number accurate to 7 decimal places.

Samples

1 5	1.000000000
5 1 1 1 1 1	5.000000000
2 1 2	2.333333333

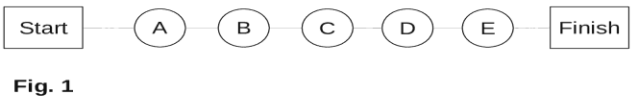


Fig. 1

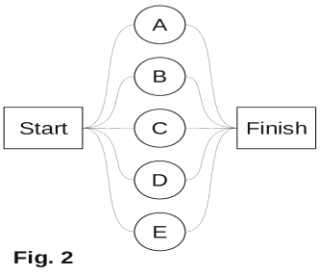


Fig. 2

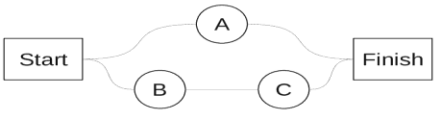


Fig. 3

Test case 1. The map can be drawn as in *Fig. 1* Any one of the stations will break the connection between start and finish so the expected value is 1.0.

Test case 2. The map can be drawn as in *Fig. 2.* If any one of the stations remain than the two cities will still be connected. Thus it will take all 5 days to disconnect the cities and the expected value is 5.0.

Test case 3. The map can be drawn as in *Fig. 3* The strike can proceed in six different ways that are equally likely, and the cities will be disconnected when A and at least one of B or C is closed. (A,B,C) (A,C,B) (B,A,C) (C,A,B) In these cases the cities are disconnected after 2 days. (B,C,A) (C,B,A) In these cases the cities are disconnected after 3 days. Thus the expected number of days is $2\frac{4}{6} + 3\frac{2}{6} = 1\frac{1}{3} + 1 = 2\frac{1}{3} = 2.333333333$

G. Sum of One-sequence

We say that a sequence of integers is a one-sequence if the difference between any two consecutive numbers in this sequence is -1 or 1 and its first element is 0. More precisely: a_1, a_2, ..., a_n is a one-sequence if:

- for any k ($1 \le k < n$): $|a_k - a_{k+1}|=1$,
- $a_1=0$.

You are given n and s — sum of all elements in a. Construct a one-sequence with the given parameters.

Input
The first line contains a pair of integers n and s ($1 \le n \le 10000$, $|s| \le 50000000$).

Output
Print required sequence or the single integer -1 if it doesn't exist.

Sample

8 4	0 1 2 1 0 -1 0 1

H. Pascal

Little Frane is already in tenth grade, but is still struggling with Pascal in computer class. For homework, his teacher wrote the following program into his notebook, and he needs to determine the output, given the integer N.

```
readln(N);
counter := 0;
for i := N-1 downto 1 do begin
    counter := counter + 1;
    if N mod i = 0 then break;
end;
writeln(counter);
```

Write a program which solves Frane's problem.

Input
The first line of input contains the integer N ($1 \le N \le 10^9$).

Output
Output the result on a single line.

Sample

1	0
10	5
27	18

I. Barica

Barica is an unusual frog. She lives in a pond where N plants float on the surface of the water. The plants are numbered 1 through N. When viewing from above, the location of each plant is given by a pair of coordinates. What makes Barica unusual is her fear of jumping diagonally and in the negative direction. More precisely, she can jump from a plant at coordinates (x_1, y_1) to another at coordinates (x_2, y_2) only if:

· $x_2 > x_1$ and $y_2 = y_1$, or
· $y_2 > y_1$ and $x_2 = x_1$

For each plant, we know the number of flies in its immediate vicinity. Barica can use her swift tongue to eat all flies near the plant she is at.

Barica absorbs one energy unit for each fly she eats, and uses K energy units for each jump she

makes. Barica can not make a jump if she doesn't have enough energy units beforehand.

Barica wants to go from plant 1 to plant N and have the largest amount of energy possible after arriving. Barica initially has no energy and must gather energy for her first jump from the flies around plant 1.

Find the sequence of plants Barica should travel to achieve her goal.

Input

The first line of input contains two integers *N* and *K* (2 ≤ *N* ≤ 300 000, 1 ≤ *K* ≤ 1000) separated by a space. Each of the following *N* lines contains three integers *X*, *Y* and *F* (0 ≤ *X*, *Y* ≤ 100000, 0 ≤ *F* ≤ 1000) separated by spaces, meaning that there is a plant at coordinates (*X*, *Y*) with *F* flies around it.

The first plant in the input is plant 1, the second is plant 2 etc. No two plants will share the same pair of coordinates.

Note: The input data will guarantee that a sequence of jumps, although not necessarily unique, will always exist.

Output

Output the final energy level on the first line. Output an integer *L*, the number of plants Barica should travel, including plants 1 and *N*. On the following *L* lines, output the sequence of plants Barica should travel.

Sample

6 5 1 1 5 2 1 5 1 2 4 2 3 5 3 2 30 3 3 5	5 4 1 1 2 1 2 3 3 3
8 10 1 1 15 2 2 30 1 2 8 2 1 7 3 2 8 2 3 7 4 2 100 3 3 15	36 5 1 1 1 2 2 2 3 2 3 3
9 5 5 5 10 6 5 2 7 5 1 5 6 2 6 6 6 7 6 2 5 7 1 6 7 2 7 7 1	2 3 5 5 7 5 7 7

J. Baza

The longest common prefix of two words is the longest word that both words start with. For example,

the longest common prefix of the words "identity" and "idealistic" is the word "ide".

A database contains *N* words. The algorithm to search for a query word *W* in the database is primitive. It compares the word *W* one by one with each word in the database. Two words are compared letter by letter until a letter in which they differ is found or until the end of one of the words is reached (it is then established either

that the words are equal or that one is longer than the other). When the algorithm finds the word W in the database, it terminates.

Analyzing the algorithm shows that the number of steps needed to find a word W is equal to the number of words W is compared to, plus the sum of the lengths of the longest common prefixes of W and each of the words it was compared to.

Write a program that calculates the number of steps the algorithm uses to find each of the Q query words.

Input

Each of the following N ($1 \le N \le 30000$) lines contains a single word from the database. The words are given in the order the algorithm compares them to a query word. All words in the database will be distinct.

The following line contains an integer Q ($1 \le Q \le 30000$), the number of words searched for.

Each of the following Q lines contains a single query word.

All words in the input will be strings of less than 30 lowercase letters of the English alphabet.

Output

Output one integer per line for each query word, the number of steps the algorithm uses when searching for the word.

Sample

5 hobotnica robot hobi hobit robi 4 robi hobi hobit rakija	12 10 16 7
8 majmunica majmun majka malina malinska malo maleni malesnica 3 krampus malnar majmun	8 29 14

K. Dependency Problems

We bought a brand new computer and now we would like to install an operating system. The only problem is that our chosen operating system consists of many packages and they cannot be installed in an arbitrary order. E.g. you cannot install the package tuxracer, which depends on the package libSDL, before you install libSDL. But libSDL can depend on another packages and so on. The packages may only be installed one at a time. You may install a package only if you already installed all

packages it depends on. Your task is to determine how many packages can be installed on our computer.

Input

The input contains a single line for each available package. The line for each package P begins with the name of the package. The name of each package is a non-empty string of printable characters containing no spaces (length doesn't exceed 100). Following the name of the package P is the dependency list of P. The dependency list is simply a list of names of packages that P depends on, separated by spaces. A whitespace followed by a single 0 (zero) is at the end of each line. You may assume that no package has the name '0'. The size of the input doesn't exceed 512 kilobytes.

Output

The output consists of one number -- the maximum number of packages that may be installed on the computer.

Sample

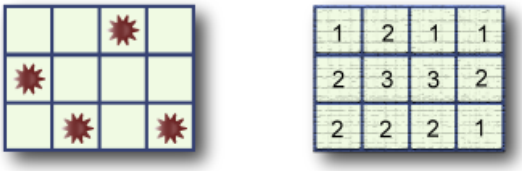
a b c b 0 b c 0 c 0 d e f 0 e f 0 f e 0 g h 0	3

L. Mine Layer

MineLayer is a MineSweeper-like puzzle game played on an R by C grid. Each square in the grid either has one mine or no mines at all. A MineLayer puzzle consists of a grid of numbers, each of which indicates the total number of mines in all adjacent squares and in the square underneath. The numbers will thus range from zero to nine.

The objective of MineLayer is to figure out a layout of the mines in the grid that matches the given clues.

Below is a typical 3 by 4 grid. The original layout is on the left, and the puzzle on the right.



Since there may be many solutions, your task is to write a program that outputs the maximum possible number of mines in the middle row. The number of rows will always be odd, and there will always be at least one solution to the puzzle.

Input

The first line of input gives the number of cases, *N* (1 ≤ *N* ≤ 50). N test cases follow.

The first line of each case contains two space-separated numbers: *R* (*R* is an odd number between 3 and 49, inclusive) the number of rows, and *C* (3 ≤ *C* ≤ 49), the number of columns. *R* is always an odd integer. Each of the next *R* lines contains *C* space-separated numbers that denote the clues of that row.

Each puzzle is guaranteed to have at least one solution.

Output

For each test case, output one line containing "Case #X: Y", where X is the 1-based case number, and Y is the maximum possible number of mines in the middle row of a grid that satisfies the given constraints.

Sample

2 3 3 2 2 1 3 4 3 2 3 2 3 4 1 2 1 1 2 3 3 2 2 2 2 1	Case #1: 1 Case #2: 1