MemSQL Start[c]UP 2.0 - Round 2

A. Golden System

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Piegirl got bored with binary, decimal and other integer based counting systems. Recently she discovered some interesting properties about number $q = \frac{\sqrt{5}+1}{2}$, in particular that $q^2 = q + 1$, and she thinks it would make a good base for her new unique system. She called it "golden system". In golden system the number is a non-empty string containing 0's and 1's as digits. The decimal value of expression $a_0a_1...a_n$ equals to $\sum_{i=0}^{n} a_i \cdot q^{n-i}$.

Soon Piegirl found out that this system doesn't have same properties that integer base systems do and some operations can not be performed on it. She wasn't able to come up with a fast way of comparing two numbers. She is asking for your help.

Given two numbers written in golden system notation, determine which of them has larger decimal value.

Input
Input consists of two lines — one for each number. Each line contains non-empty string consisting of '0' and '1' characters. The length of each string does not exceed 100000.

Output
Print ">" if the first number is larger, "<" if it is smaller and "=" if they are equal.

Examples

input
1000 111
output
<

input
00100 11
output
=

input
110 101
output
>

Note
In the first example first number equals to $((\sqrt{5}+1)/2)^3 \approx 1.618033988^3 \approx 4.236$, while second number is approximately $1.618033988^2 + 1.618033988 + 1 \approx 5.236$, which is clearly a bigger number.

In the second example numbers are equal. Each of them is ≈ 2.618.

B. Distributed Join

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Piegirl was asked to implement two table join operation for distributed database system, minimizing the network traffic.

Suppose she wants to join two tables, A and B. Each of them has certain number of rows which are distributed on different number of partitions. Table A is distributed on the first cluster consisting of m partitions. Partition with index i has a_i rows from A. Similarly, second cluster containing table B has n partitions, i-th one having b_i rows from B.

In one network operation she can copy one row from any partition to any other partition. At the end, for each row from A and each row from B there should be a partition that has both rows. Determine the minimal number of network operations to achieve this.

Input

First line contains two integer numbers, m and n ($1 \le m, n \le 10^5$). Second line contains description of the first cluster with m space separated integers, a_i ($1 \le a_i \le 10^9$). Similarly, third line describes second cluster with n space separated integers, b_i ($1 \le b_i \le 10^9$).

Output

Print one integer — minimal number of copy operations.

Examples

input
2 2 2 6 3 100
output
11

input
2 3 10 10 1 1 1
output
6

Note

In the first example it makes sense to move all the rows to the second partition of the second cluster which is achieved in $2 + 6 + 3 = 11$ operations

In the second example Piegirl can copy each row from B to the both partitions of the first cluster which needs $2 \cdot 3 = 6$ copy operations.

C. Elections

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are running for a governor in a small city in Russia. You ran some polls and did some research, and for every person in the city you know whom he will vote for, and how much it will cost to bribe that person to vote for you instead of whomever he wants to vote for right now. You are curious, what is the smallest amount of money you need to spend on bribing to win the elections. To win elections you need to have strictly more votes than any other candidate.

Input

First line contains one integer n ($1 \le n \le 10^5$) — number of voters in the city. Each of the next n lines describes one voter and contains two integers a_i and b_i ($0 \le a_i \le 10^5$; $0 \le b_i \le 10^4$) — number of the candidate that voter is going to vote for and amount of money you need to pay him to change his mind. You are the candidate 0 (so if a voter wants to vote for you, a_i is equal to zero, in which case b_i will also be equal to zero).

Output

Print one integer — smallest amount of money you need to spend to win the elections.

Examples

input
5 1 2 1 2 1 2 2 1 0 0
output
3

input
4 1 2 1 2 2 1 0 0
output
2

input
1 100000 0
output
0

D. Bingo!

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The game of bingo is played on a 5×5 square grid filled with distinct numbers between 1 and 75. In this problem you will consider a generalized version played on an $n \times n$ grid with distinct numbers between 1 and m ($m \geq n^2$).

A player begins by selecting a randomly generated bingo grid (generated uniformly among all available grids). Then k distinct numbers between 1 and m will be called at random (called uniformly among all available sets of k numbers). For each called number that appears on the grid, the player marks that cell. The score at the end is 2 raised to the power of (number of completely marked rows plus number of completely marked columns).

Determine the expected value of the score. The expected score may be very large. If the expected score is larger than 10^{99}, print 10^{99} instead (for example as "1e99" without the quotes).

Input

Input will consist of three integers n, m, k ($1 \leq n \leq 300$; $n^2 \leq m \leq 100000$; $n \leq k \leq m$).

Output

Print the smaller of 10^{99} and the expected score. Your answer must be correct within an absolute or relative error of 10^{-9}.

Examples

input
1 2 1
output
2.5

input
2 4 3
output
4

input
7 59164 40872
output
3.1415926538

E. Flow Optimality

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

There is a computer network consisting of n nodes numbered 1 through n. There are links in the network that connect pairs of nodes. A pair of nodes may have multiple links between them, but no node has a link to itself.

Each link supports unlimited bandwidth (in either direction), however a link may only transmit in a single direction at any given time. The cost of sending data across a link is proportional to the square of the bandwidth. Specifically, each link has a positive weight, and the cost of sending data across the link is the weight times the square of the bandwidth.

The network is connected (there is a series of links from any node to any other node), and furthermore designed to remain connected in the event of any single node failure.

You needed to send data from node 1 to node n at a bandwidth of some positive number k. That is, you wish to assign a bandwidth to each link so that the bandwidth into a node minus the bandwidth out of a node is $-k$ for node 1, k for node n, and 0 for all other nodes. The individual bandwidths do not need to be integers.

Wishing to minimize the total cost, you drew a diagram of the network, then gave the task to an intern to solve. The intern claimed to have solved the task and written the optimal bandwidths on your diagram, but then spilled coffee on it, rendering much of it unreadable (including parts of the original diagram, and the value of k).

From the information available, determine if the intern's solution may have been optimal. That is, determine if there exists a valid network, total bandwidth, and optimal solution which is a superset of the given information. Furthermore, determine the efficiency of the intern's solution (if possible), where efficiency is defined as total cost divided by total bandwidth.

Input

Input will begin with two integers n and m ($2 \le n \le 200000$; $0 \le m \le 200000$), the number of nodes and number of known links in the network, respectively. Following this are m lines with four integers each: f, t, w, b ($1 \le f \le n$; $1 \le t \le n$; $f \ne t$; $1 \le w \le 100$; $0 \le b \le 100$). This indicates there is a link between nodes f and t with weight w and carrying b bandwidth. The direction of bandwidth is from f to t.

Output

If the intern's solution is definitely not optimal, print "BAD x", where x is the first link in the input that violates the optimality of the solution. If the intern's solution may be optimal, print the efficiency of the solution if it can be determined rounded to the nearest integer, otherwise print "UNKNOWN".

Examples

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

input
5 5 2 3 1 1 3 4 1 1 4 2 1 1 1 5 1 1 1 5 100 100
output
BAD 3

input
6 4 1 3 31 41 1 5 59 26 2 6 53 58 4 6 97 93
output
UNKNOWN

input
7 5 1 7 2 1 2 3 1 1 4 5 1 0 6 1 10 0 1 3 1 1

output
BAD 4

Note

Although the known weights and bandwidths happen to always be integers, the weights and bandwidths of the remaining links are not restricted to integers.

F. An easy problem about trees

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pieguy and Piegirl are playing a game. They have a rooted binary tree, that has a property that each node is either a leaf or has exactly two children. Each leaf has a number associated with it.

On his/her turn a player can choose any two leafs that share their immediate parent, remove them, and associate either of their values with their parent, that now became a leaf (the player decides which of the two values to associate). The game ends when only one node (the one that was the root of the tree) is left.

Pieguy goes first, and his goal is to maximize the value that will be associated with the root when the game ends. Piegirl wants to minimize that value. Assuming that both players are playing optimally, what number will be associated with the root when the game ends?

Input

First line contains a single integer t ($1 \le t \le 100$) — number of test cases. Then t test cases follow. Each test case begins with an empty line, followed by a line with a single integer n ($1 \le n \le 250$), followed by n lines describing n nodes of the tree. Each of those n lines either contains a non-negative number a_i, indicating a leaf node with value a_i ($0 \le a_i \le 1000$) associated with it, or -1 followed by integers l and r, indicating a non-leaf node with children l and r ($0 \le l, r \le n - 1$). Nodes are numbered from 0 to $n - 1$. The root is always node 0.

Output

For each test case print one line with one integer on it — the number that will be associated with the root when the game ends.

Examples

input
4 3 -1 1 2 10 5 5 -1 1 2 -1 3 4 10 5 20 7 -1 1 2 -1 3 4 -1 5 6 1 2 3 4 11 -1 1 2 -1 3 4 -1 5 6 -1 7 8 15 7 -1 9 10 7 8 9 11

output
10 10 4 8