Problem A. Absolute Difference

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Both Alice and Bob have a set of real numbers, and both sets are the union of some disjoint closed intervals. They will independently pick a real number uniformly at random from their own set, and you need to calculate the expected absolute difference between the two real numbers.

More formally, given a set of real numbers $S = \bigcup [l, r]$, picking a real number x from the set S uniformly at random means that $P(x \in [l_1, r_1]) = P(x \in [l_2, r_2])$ holds for any two intervals $[l_1, r_1], [l_2, r_2] \subseteq S$ with the same length, i.e., $r_1 - l_1 = r_2 - l_2$.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$), the number of intervals that form Alice's set and Bob's set respectively.

Each of the following $n + m$ lines contains two integers l and r ($-10^9 \le l \le r \le 10^9$), describing a closed interval $[l, r]$. The first n intervals form Alice's set and the next m intervals form Bob's set. Note that an interval $[l, r]$ with $l = r$ is a degenerate interval that contains a single real number.

It is guaranteed that the intervals that form someone's set are pairwise disjoint.

Output

Output a single real number, indicating the expected absolute difference of the two real numbers picked by Alice and Bob separately.

Your answer is acceptable if its absolute or relative error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$.

Examples

standard input	standard output
1 1 0 1 0 1	0.333333333333333
1 1 0 1 1 1	0.5

Note

In the first sample case, both Alice and Bob can pick any real number from $[0, 1]$, and the expected absolute difference is $\int_0^1 \int_0^1 |x - y| \, dx \, dy = \frac{1}{3}$.

In the second sample case, Alice can pick any real number from $[0, 1]$ while Bob can only pick 1, and therefore the expected absolute difference is $\int_0^1 |x - 1| \, dx = \frac{1}{2}$.

Problem B. Compiler

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	1 second
Memory limit:	512 mebibytes

One thing almost all computers have in common, whether it is a simple stack-based calculator, a 6502 powered BBC Micro, or a brand new Android phone, every modern computer requires programs that turn either high-level languages or assembly language into machine code.

UKIEPC recently designed their own processor. While very easy to load programs onto, it is not as complex as some, since it has only one aim: show a number on an innovative laser display board!

The available memory consists of three 8-bit registers: `A`, `X`, and `Y`, plus an infinite stack. At program start these registers are initialised to unknown values, and the stack is empty.

The processor supports six unique instructions:

- `PH <reg>`: push the contents of the register (i.e. `A`, `X`, or `Y`) onto the stack.
- `PL <reg>`: pop a value off the stack into the register. The program will terminate if this instruction is called when the stack is empty.
- `AD`: pop two values off the stack, and push the lowest 8 bits of their sum onto the stack.
- `ZE <reg>`: set the register equal to zero.
- `ST <reg>`: set the register equal to one.
- `DI <reg>`: send the value of the register to the laser display board and exit.

Due to memory constraints, the maximum number of instructions that can be written to disk is 40. Further instructions will not be executed.

Given a number, write a program to output the number on the laser display board.

Input

- One line containing an integer N ($0 \le N \le 255$), the number to output.

Output

- At most 40 lines, each containing one valid instruction.

  When run in sequence the lines should output the number N. The last instruction should be a `DI`.

Examples

input.txt	output.txt
2	ST A ST X PH A PH X AD PL Y DI Y
5	ST X PH X PH X PH X AD PL Y PH Y PH Y AD AD PL A DI A

Problem C. Clamped Sequence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

Given an integer sequence $a_1, a_2, \ldots, a_n$ and a positive integer d, you need to clamp the sequence to a range $[l, r]$ satisfying $0 \le r - l \le d$ that maximize $\sum_{i=1}^{n-1} |a_i - a_{i+1}|$, where $|x|$ is the absolute value of x.

More specifically, clamping the sequence to the range $[l, r]$ makes each element

$$a_i := \begin{cases} l, & a_i < l; \\ a_i, & l \le a_i \le r; \\ r, & a_i > r. \end{cases}$$

Both l and r are arbitrary real numbers decided by you under the given constraints. It can be shown that the maximum sum is always an integer.

Input

The first line contains two integers n ($2 \le n \le 5\,000$) and d ($1 \le d \le 10^9$), denoting the length of the given sequence and the given parameter respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$), denoting the given sequence.

Output

Output a line containing a single integer, denoting the maximum sum.

Example

standard input	standard output
8 3 3 1 4 1 5 9 2 6	15

Note

In the sample case, you can clamp the given sequence to the range $[1, 4]$ to make it $[3, 1, 4, 1, 4, 4, 2, 4]$, and the resulting sum is the maximum 15.

Problem D. DRX vs. T1

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	512 megabytes

The League of Legends Worlds Championship, Worlds for short, is the culminating event in the LoL Esports season and decides the ultimate global League of Legends Champion.

Worlds 2022 has been a great tournament thus far. We have seen the legend, Faker, rise once more and the surprise of the DRX run. There have also been massive collapses from other teams like Gen.G and JDG. Regardless, we are in for a compelling Worlds 2022 Finals, which kicks off on November 6 at 08:00 (UTC+8) in San Francisco as DRX and T1 face off.

Being the clear favorite for winning the Finals, T1 has performed dominantly in most games and has found its stride coming into the tournament. They have taken down RNG and JDG, two LPL heavyweights, and are now poised to add another trophy to their cabinet.

On the other side, DRX has practically come out of nowhere. From almost not making Worlds 2022, fighting their way through the Regionals and Play-ins to the Finals. It is a Cinderella run that DRX is on, and they have been doubted at every step of the tournament. With a lot to prove and Deft reaching the Finals in his swan song, winning the trophy might be the best send-off any player can wish for.

As we countdown to a battle for the all-new trophy, you will be given the predicted result of the Finals and required to find which team wins the championship under the prediction. A team must win three out of five games to win the championship, so the predicted result of the Finals is a string of length 5 containing only Ds, Ts, and ?s. The i-th character in the string points out who wins the i-th game, where D means DRX wins the game, T means T1 wins the game, and ? means no more games are needed since some team has already won 3 games and become the champion.

Input

The only line contains a single string of length 5 containing only Ds, Ts, and ?s, indicating the predicted result of the Worlds 2022 Finals.

It is guaranteed that the predicted result is a valid BO5 result, that is, no ?s are before any Ds and Ts, and there exists a team who wins exactly three games.

Output

Output a line containing a single string, indicating the team that wins the champion under the prediction. That is, if DRX wins the champion, output "`DRX`" (without quotes), and if T1 wins, output "`T1`" (without quotes).

Examples

standard input	standard output
`TDTT?`	`T1`
`DTDD?`	`DRX`

Note

In the first sample case, T1 wins the first game, DRX wins the second game, T1 wins the third and the fourth game and wins the champion, so the fifth game is not needed anymore.

Problem E. Rhyming Slang

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	1 second
Memory limit:	512 mebibytes

Rhyming slang involves replacing a common word with a phrase of several (really two or three) words, the last of which rhymes with the original word. For example,

- replacing the word “stairs” with the rhyming phrase “apples and pears”,
- or replacing “rotten” with the phrase “bales of cotton”.

English has such a wide variety of spellings and pronunciations that for any non-native speaker telling what rhymes isnвЂ™t always easy. Perhaps you can help?

Typically, two words rhyme (or can be forced to rhyme) if both of their endings can be found on the same list of word endings that sound the same.

Given a common word, a number of lists, each containing word endings that sound the same, and a number of phrases, determine if those phrases could be rhyming slang.

Input

- One line containing the single common word S ($1 \le |S| \le 20$).
- One line containing an integer E ($1 \le E \le 10$), the number of lists of word endings that sound the same.
- E lines, each no more than 100 characters long. Each a list of space-separated word endings.
- One line containing an integer P ($1 \le P \le 10$), the number of phrases to test.
- P lines, each no more than 100 characters long, containing a phrase p_i of at least one and no more than nine words that might rhyme with the common word.

All words and letters will be in lower case. The common word’s ending will appear in at least one ending list.

Output

- P lines, each consisting of either:
  - ’`YES`’: The phrase p_i rhymes with the common word.
  - ’`NO`’: The phrase p_i does not rhyme with the common word.

Examples

<table>
<tr><th>input.txt</th><th>output.txt</th></tr>
<tr><td>stairs
2
erres airs ears ares aires
eet eat
2
apples and pears
plates of meat</td><td>YES
NO</td></tr>
<tr><td>drought
2
aught ought aut acht
ought oubt outte out oute
5
tasty sprout
difficult route
worried and fraught
forever in doubt
apples and pears</td><td>YES
YES
YES
YES
NO</td></tr>
<tr><td>listen
1
issen isten ison
2
glisten
listen</td><td>YES
YES</td></tr>
</table>

Problem F. Half Mixed

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

Given two integers n and m, you need to construct a matrix M satisfying the following constraints:

- The number of rows and columns of M are n and m respectively.
- The matrix contains only 0s and 1s, namely, $M_{i,j} \in \{0, 1\}$ for all $1 \le i \le n$ and $1 \le j \le m$.
- The number of mixed subrectangles equals to the number of pure subrectangles. The subrectangles that contain both 0s and 1s are considered as mixed subrectangles, otherwise pure subrectangles. Note that a subrectangle is an intersection of some consecutive rows and some consecutive columns.

If multiple solutions exist, print any one of them. If there is no solution, report it.

Input

The first line contains an integer T ($1 \le T \le 10^5$), denoting the number of test cases.

For each test case, the only line contains two integers n and m ($1 \le n, m \le 10^6$, $1 \le n \times m \le 10^6$), denoting the number of rows and columns respectively.

It is guaranteed that the sum of $n \times m$ over all test cases does not exceed 5×10^6.

Output

For each test case, if there is no solution, print "`No`" (without quotes) in one line. If the solution exists, print "`Yes`" (without quotes) in the first line. Then print n lines, the i-th of which contains m integers $M_{i,1}, M_{i,2}, \ldots, M_{i,m}$, describing the i-th row of the matrix.

Example

standard input	standard output
2 2 3 1 1	Yes 0 1 1 1 1 0 No

Note

In the first sample case, the number of mixed subrectangles and pure subrectangles are both 9.

In the second sample case, the only subrectangle is the whole matrix that must be pure, so the number of mixed subrectangles and the number of pure subrectangles must be 0 and 1 respectively, which are not equal.

Problem G. Meet in the Middle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	512 megabytes

Byteland has n cities numbered from 1 to n, and $n-1$ bidirectional roads and $n-1$ bidirectional railways connecting them. For each pair of cities, one can always arrive one from another one through the roads only, and the same thing also holds for the railways.

Alice and Bob are planning q long journeys in Byteland. In each journey, Alice starts from the a-th city and then visits some other cities through the roads only, while Bob starts from the b-th city and then visits some other cities through the railways only. Both of them will finally reach the **same** destination city without visiting any cities more than once. You need to find the maximum total length of their journey routes among all the selections of the destination city.

Input

The first line contains two integers n ($2 \le n \le 10^5$) and q ($1 \le q \le 5 \times 10^5$), indicating the number of cities in Byteland and the number of journey plans.

Each of the next $n-1$ lines contains three integers u, v ($1 \le u, v \le n$) and w ($1 \le w \le 10^9$), indicating a road of length w connecting the u-th and v-th cities.

Each of the next $n-1$ lines contains three integers u, v ($1 \le u, v \le n$) and w ($1 \le w \le 10^9$), indicating a railway of length w connecting the u-th and v-th cities.

Each of the next q lines contains two integers a and b ($1 \le a, b \le n$), indicating a journey where Alice starts from the a-th city and Bob starts from the b-th city.

Output

Output q lines, each of which contains a single integer, indicating the maximum total length of Alice's and Bob's journey routes among all the selections of the destination city.

Example

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

Problem H. P-P-Palindrome

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

Given n strings $S_1, S_2, \ldots S_n$, you need to calculate the number of different *P-P-Palindromes* given by these n strings.

A palindrome is a string that can be read the same from left to right and from right to left. For example, "`a`", "`level`", and "`otto`" are palindromes, while "`aab`" and "`icpc`" are not.

A *P-P-Palindrome* is an ordered pair of **nonempty** palindromes (P, Q) such that both P and Q are the substrings of some in $S_1, S_2, \ldots S_n$ and $P + Q$ is also a palindrome. Here $P + Q$ means concatenating P and Q in order, or more specifically, let $P = p_1 p_2 \ldots p_{|P|}$ and $Q = q_1 q_2 \ldots q_{|Q|}$, then $P + Q = p_1 p_2 \ldots p_{|P|} q_1 q_2 \ldots q_{|Q|}$, where $|S|$ is the length of string S.

Note that two *P-P-Palindromes* are considered different if and only if P differs or Q differs.

Input

The first line contains an integer n ($1 \le n \le 10^6$), indicating the number of given strings.

Then n lines follow, the i-th of which contains a string S_i ($1 \le |S_i| \le 10^6$) consisting of lowercase English letters only.

It is guaranteed that the total length of the given strings does not exceed 10^6.

Output

Output a line containing a single integer, indicating the number of different *P-P-Palindromes* given by the n strings.

Examples

standard input	standard output
2 aaaa aaa	16
3 abaaa abbbba bbbaba	28

Problem I. Quartz Collection

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	512 megabytes

There is a shop selling n types of quartz in Byteland. Only two pieces of each type of quartz will be on sale every day, and the second piece will be on sale **only after** the first piece has been sold.

Two wizards Alice and Bob are collecting these n types of quartz to strengthen their wand. Due to the quartz shortage, they reach an agreement that either of them buy only one piece of each type each day.

Both of them wants to minimize their own cost each day. To reflect fairness, Alice buys one piece of quartz first, and then Bob and Alice buy two pieces in turn until only one piece remains. The one who still hasn't collected all types of quartz buys the last piece.

In each of the following m days, the prices of the two pieces of one type of quartz will change **permanently**, and Alice wants to know the minimum cost for her to collect all types of quartz if both Alice and Bob take the best strategy on the first day and each of the following m days.

Input

The first line contains two integers n and m ($1 \le n, m \le 10^5$), indicating the number of types of quartz and the number of following days.

Then n lines follow, the i-th of which contains two integers a and b ($1 \le a, b \le 10^5$), indicating the prices of the first piece and the second piece of the i-th type of quartz on sale respectively.

Then m lines follow, each line contains three integers t ($1 \le t \le n$), x and y ($1 \le x, y \le 10^5$), indicating that the prices of the first piece and the second piece of the t-th type of quartz on sale will change to x and y respectively.

Output

Output $m + 1$ lines, each containing a single integer, indicating the minimum cost for Alice to collect all types of quartz on the first day and each of the following m days.

Example

standard input	standard output
4 5 2 4 5 7 1 7 2 1 4 5 2 1 6 2 4 4 3 2 1 3 3 6 6	13 14 15 14 10 13

Note

In the sample case, one of the best strategies on the first day is as follows:

- Alice buys the first piece of the third type of quartz.
- Bob buys the first piece of the first type and the second type.
- Alice buys the second piece of the first type and the second type.
- Bob buys the second piece of the third type and the first piece of the fourth type.
- Alice buys the second piece of the fourth type, which is also the last piece of quartz.

Problem J. Grass Seed Inc.

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	1 second
Memory limit:	512 mebibytes

Many years ago after another unfruitful day in Cubicle Land, banging her head against yet another cutting edge, marketing buzzword-filled JavaScript framework, Janice the engineer looked out of the window and decided that time was ripe for a change.

So swapping her keyboard and mouse for a fork and a spade, she started her own gardening company.

After years of hard outdoor work Janice now has biceps like Van Damme and owns the premiere landscaping company in the whole of the South West, and has just been lucky enough to broker a large contract to sow lawns for landed gentry.

Each contract details the size of the lawns that need to be seeded, and the cost of seed per square metre. How much do you need to spend on seed?

Input

- One line containing a floating point number C ($0 < C \leq 100$), the cost of seed to sow one square metre of lawn.
- One line containing an integer L ($0 < L \leq 100$), the number of lawns to sow.
- L lines, each containing two positive floating point numbers: w_i ($0 \leq w_i \leq 100$), the width of the lawn, and l_i ($0 \leq l_i \leq 100$), the length of the lawn.

Output

- One line containing a real number: the cost to sow all of the lawns.

All output must be accurate to an absolute or relative error of at most 10^{-6}.

Examples

input.txt	**output.txt**
2 3 2 3 4 5 5 6	112.0000000
0.75 2 2 3.333 3.41 4.567	16.6796025

Problem K. Secret Santa

Input file:	`input.txt`
Output file:	`output.txt`
Time limit:	1 second
Memory limit:	512 mebibytes

Christmas comes sooner every year. In fact, in one oft-forgotten corner of the world, gift-giving has already started in the form of a *Secret Santa* syndicate.

Everybody in the small town of Haircombe is going to put their name into a hat. This hat will be given a hearty shuffle, and then afterwards everybody will take turns once more in taking a name back from the hat.

The name each person receives is the name of the fellow citizen to whom they will send a gift.

Of course, one concern with this strategy is that some unfortunate citizens could wind up giving gifts to themselves. What are the chances that this will happen to any of the citizens of Haircombe?

Input

- One line containing the number N ($1 \le N \le 10^{12}$), the number of citizens who will take part in Secret Santa.

Output

- One line containing one real number; the probability that one or more people wind up giving gifts to themselves.

All output must be accurate to an absolute or relative error of at most 10^{-6}.

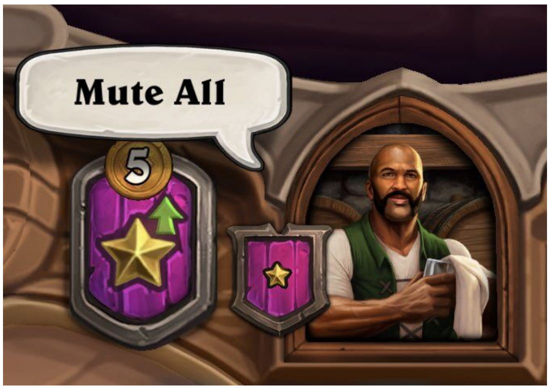
Examples

input.txt	output.txt
2	0.50000000
3	0.66666667
6	0.63194444

Problem L. Tavern Chess

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

One day, Alice and Bob are playing an online chess game named "Tavern Chess", where each player can choose at most 7 minions to build a team.



Welcome to Tavern

Each minion has its Hit Points (HP) and Attack (ATK), and **the HP is the same as the ATK initially**. A minion with positive HP is alive and can take attacks, and it dies immediately if its HP becomes zero or lower after an attack.

The battle begins after both Alice and Bob finish building their team and lasts until all the minions from one team are dead and the other team wins. In case the last alive minions from the two teams die simultaneously, the battle ends in a tie.

Alice and Bob's teams take turns to attack, and the team that has more minions takes the first attack. In case of a tie, it will be decided by a coin toss — 50% probability for Alice taking the first attack and the remaining 50% probability for Bob taking the first attack.

When a team takes the attack, **the leftmost minion with the minimum number of taking attacks** from the team attacks one of the alive minions from the other team uniformly at random, and then the other team takes the attack.

If a minion with the HP h_1 and the ATK a_1 attacks another minion with the HP h_2 and the ATK a_2, each minion deals damages equal to its ATK to the other one, so the attacker minion will have the HP $h_1 - a_2$ and the attacked minion will have the HP $h_2 - a_1$ after the attack.

Given Alice's team and Bob's team, you need to calculate the probabilities that Alice's team wins the battle, Bob's team wins the battle, or the battle ends in a tie, respectively.

Input

The first line contains two integers n and m ($1 \le n, m \le 7$), indicating the number of minions in Alice's team and Bob's team respectively.

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$), the i-th of which is the HP as well as the ATK of the i-th minions from the left from Alice's team.

The third line contains m integers $b_1, b_2, \ldots, b_m$ ($1 \le b_i \le 10^9$), the i-th of which is the HP as well as the ATK of the i-th minions from the left from Bob's team.

Output

Output three lines, each containing a single real number, indicating the probabilities that Alice's team wins the battle, Bob's team wins the battle, or the battle ends in a tie, respectively.

Your answer is acceptable if its absolute error does not exceed 10^{-9}. Formally speaking, suppose that your output is a and the jury's answer is b, your output is accepted if and only if $|a - b| \le 10^{-9}$.

Examples

standard input	standard output
2 3 2 5 3 4 1	0.125 0.75 0.125
6 6 1 1 4 5 1 4 1 1 4 5 1 4	0.241867283950617 0.241867283950617 0.516265432098765