Problem A. Accounting

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

A long time ago in some far country lived king Copa. After the recent king's reform, he got so large powers that started to keep the books by himself.

The total income A of his kingdom during 0-th year is known, as well as the total income B during n-th year (these numbers can be negative — it means that there was a loss in the correspondent year).

King wants to show financial stability. To do this, he needs to find common coefficient X — the coefficient of income growth during one year. This coefficient should satisfy the equation:

$$A \cdot X^n = B.$$

Surely, the king is not going to do this job by himself, and demands you to find such number X.

It is necessary to point out that the fractional numbers are not used in kingdom's economy. That's why all input numbers as well as coefficient X must be integers. The number X may be zero or negative.

Input

The input contains three integers A, B, n ($|A|, |B| \leq 1000$, $1 \leq n \leq 10$).

Output

Output the required integer coefficient X, or «No solution», if such a coefficient does not exist or it is fractional. If there are several possible solutions, output any of them.

Examples

stdin	stdout
`2 18 2`	`3`
`-1 8 3`	`-2`
`0 0 10`	`5`
`1 16 5`	`No solution`

Problem B. Codeforces World Finals

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

The king Copa often has been reported about the Codeforces site, which is rapidly getting more and more popular among the brightest minds of the humanity, who are using it for training and competing. Recently Copa understood that to conquer the world he needs to organize the world Codeforces tournament. He hopes that after it the brightest minds will become his subordinates, and the toughest part of conquering the world will be completed.

The final round of the Codeforces World Finals 20YY is scheduled for $DD.MM.YY$, where DD is the day of the round, MM is the month and YY are the last two digits of the year. Bob is lucky to be the first finalist form Berland. But there is one problem: according to the rules of the competition, all participants must be at least 18 years old at the moment of the finals. Bob was born on $BD.BM.BY$. This date is recorded in his passport, the copy of which he has already mailed to the organizers. But Bob learned that in different countries the way, in which the dates are written, differs. For example, in the US the month is written first, then the day and finally the year. Bob wonders if it is possible to rearrange the numbers in his date of birth so that he will be at least 18 years old on the day $DD.MM.YY$. He can always tell that in his motherland dates are written differently. Help him.

According to another strange rule, eligible participant must be born in the same century as the date of the finals. If the day of the finals is participant's 18-th birthday, he is allowed to participate.

As we are considering only the years from 2001 to 2099 for the year of the finals, use the following rule: the year is leap if it's number is divisible by four.

Input

The first line contains the date $DD.MM.YY$, the second line contains the date $BD.BM.BY$. It is guaranteed that both dates are correct, and YY and BY are always in $[01; 99]$.

It could be that by passport Bob was born after the finals. In this case, he can still change the order of numbers in date.

Output

If it is possible to rearrange the numbers in the date of birth so that Bob will be at least 18 years old on the $DD.MM.YY$, output `YES`. In the other case, output `NO`.

Each number contains exactly two digits and stands for day, month or year in a date. Note that it is permitted to rearrange only numbers, not digits.

Examples

stdin	stdout

`01.01.98`
`01.01.80`
	`YES`

`20.10.20`
`10.02.30`
	`NO`

`28.02.74`
`28.02.64`
	`NO`

Problem C. Shooting Gallery

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

One warm and sunny day king Copa decided to visit the shooting gallery, located at the Central Park, and try to win the main prize — big pink plush panda. The king is not good at shooting, so he invited you to help him.

The shooting gallery is an infinite vertical plane with Cartesian coordinate system on it. The targets are points on this plane. Each target is described by it's coordinates x_i, and y_i, by the time of it's appearance t_i and by the number p_i, which gives the probability that Copa hits this target if he aims at it.

A target appears and disappears instantly, so Copa can hit the target only if at the moment t_i his gun sight aimed at (x_i, y_i). Speed of movement of the gun sight on the plane is equal to 1. Copa knows all the information about the targets beforehand (remember, he is a king!). He wants to play in the optimal way, which maximizes the expected value of the amount of hit targets. He can aim at any target at the moment 0.

Input

The first line contains integer n ($1 \le n \le 1000$) — amount of targets in the shooting gallery. Then n lines follow, each describing one target. Each description consists of four numbers x_i, y_i, t_i, p_i (where x_i, y_i, t_i — integers, $-1000 \le x_i, y_i \le 1000, 0 \le t_i \le 10^9$, real number p_i is given with no more than 6 digits after the decimal point, $0 \le p_i \le 1$). No two targets may be at the same point.

Output

Output the maximum expected value of the amount of targets that was shot by the king. Your answer will be accepted if it differs from the correct answer by not more than 10^{-6}.

Examples

stdin	stdout

`1`
`0 0 0 0.5`
	`0.5000000000`

`2`
`0 0 0 0.6`
`5 0 5 0.7`
	`1.3000000000`

Problem D. King's Problem?

Input file:	`stdin`
Output file:	`stdout`
Time limit:	3 seconds
Memory limit:	256 megabytes

Every true king during his life must conquer the world, hold the Codeforces world finals, win pink panda in the shooting gallery and travel all over his kingdom.

King Copa has already done the first three things. Now he just needs to travel all over the kingdom. The kingdom is an infinite plane with Cartesian coordinate system on it. Every city is a point on this plane. There are n cities in the kingdom at points with coordinates $(x_1, 0), (x_2, 0), \cdots, (x_n, 0)$, and there is one city at point (x_{n+1}, y_{n+1}).

King starts his journey in the city number k. Your task is to find such route for the king, which visits all cities (in any order) and has minimum possible length. It is allowed to visit a city twice. The king can end his journey in any city. Between any pair of cities there is a direct road with length equal to the distance between the corresponding points.

No two cities may be located at the same point.

Input

The first line contains two integers n and k ($1 \le n \le 10^5, 1 \le k \le n+1$) — amount of cities and index of the starting city. The second line contains $n+1$ numbers x_i. The third line contains y_{n+1}. All coordinates are integers and do not exceed 10^6 by absolute value. No two cities coincide.

Output

Output the minimum possible length of the journey. Your answer must have relative or absolute error less than 10^{-6}.

Examples

stdin	stdout

`3 1`
`0 1 2 1`
`1`
	`3.41421356237309490000`

`3 1`
`1 0 2 1`
`1`
	`3.82842712474619030000`

`4 5`
`0 5 -1 -5 2`
`3`
	`14.24264068711928400000`

Problem E. Tricky and Clever Password

Input file:	`stdin`
Output file:	`stdout`
Time limit:	2 seconds
Memory limit:	256 megabytes

In his very young years the hero of our story, king Copa, decided that his private data was hidden not enough securely, what is unacceptable for the king. That's why he invented tricky and clever password (later he learned that his password is a palindrome of odd length), and coded all his data using it.

Copa is afraid to forget his password, so he decided to write it on a piece of paper. He is aware that it is insecure to keep password in such way, so he decided to cipher it the following way: he cut x characters from the start of his password and from the end of it (x can be 0, and $2x$ is strictly less than the password length). He obtained 3 ***parts of the password***. Let's call it $prefix$, $middle$ and $suffix$ correspondingly, both $prefix$ and $suffix$ having equal length and $middle$ always having odd length. From these parts he made a string $A + prefix + B + middle + C + suffix$, where A, B and C are some (possibly empty) strings invented by Copa, and «+» means concatenation.

Many years have passed, and just yesterday the king Copa found the piece of paper where his ciphered password was written. The password, as well as the strings A, B and C, was completely forgotten by Copa, so he asks you to find a password of maximum possible length, which could be invented, ciphered and written by Copa.

Input

The input contains single string of small Latin letters with length from 1 to 10^5 characters.

Output

The first line should contain integer k — amount of ***nonempty parts of the password*** in your answer ($k \in \{1, 3\}$). In each of the following k lines output two integers x_i and l_i — start and length of the corresponding part of the password. Output pairs in order of increasing x_i. Separate the numbers in pairs by a single space.

Starting position x_i should be an integer from 1 to the length of the input string. All l_i must be positive, because you should output only non-empty parts. The middle part must have odd length.

If there are several solutions, output any. Note that your goal is to maximize the sum of l_i, but not to maximize k.

Examples

stdin	stdout

`abacaba`	`1`
	`1 7`

`axbya`	`3`
	`1 1`
	`2 1`
	`5 1`

`xabyczba`	`3`
	`2 2`
	`4 1`
	`7 2`