# Problem A. Alphabet

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Little Kostya started to learn alphabet a few weeks ago, so his mom bought him a set of cards. Each of those cards contains a picture and a related word.

Kostya wants to distribute cards in piles in such way that the first letter for all words in the pile will be the same. Help him to do it.

## Input

First line of the input contains one integer $N$ $(1 \leq N \leq 100)$ — number of cards Kostya gets. Each of the next $N$ line contains one non-empty word, consisting of no more than 20 lowercase English letters. It is guaranteed that all words are pairwise different.

## Output

In the first line of the input print one integer $K$ — the number of piles Kostya gets at the end of process.

Then print $K$ lines. Each line must contain space-separated list of words, written on the cards in next pile. Piles and words inside one line must be sorted lexicographically.

## Examples

| standard input | standard output |
|---|---|
| 4<br>apple<br>cat<br>dog<br>angel | 3<br>angel apple<br>cat<br>dog |
| 3<br>he<br>she<br>it | 3<br>he<br>it<br>she |

# Problem B. Fractions

| | |
|---|---|
| Input file: | `stdin` |
| Output file: | `stdout` |
| Time limit: | 1 second |
| Memory limit: | 16 megabytes |

Father told Little Kostya about the irreducible fractions. Fraction is called irreducible if the greatest common divisor of numerator and denominator is equal to 1.

Find number of proper irreducible fractions not exceeding $X$ with denominator not exceeding $N$. Mind that fraction is called *proper*, if its numerator and denominator are positive integers and numerator is strictly less than denominator.

## Input

First line of the input contains value of $X$ ($0 < X \leq 1$) given with no more than 4 digits after the period. Second line contains one integer $N$ ($2 \leq N \leq 10^4$).

## Output

Print one integer — number of the fractions you have found.

## Examples

| stdin | stdout |
|---|---|
| 0.4<br>6 | 5 |
| 0.6543<br>10 | 20 |

## Note

The first sample list of fractions consists of 1/6, 1/5, 1/4, 1/3 and 2/5.

# Problem C. Geometry exercise

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Little Kostya studies not only numbers, but also other sections of math and informatics. While studying geometry he often draws different shapes on the squared paper using a ruler and compasses.

First he drew a rectangle. The coordinates of its opposite corners are $(x_1, y_1)$ and $(x_2, y_2)$. Afterwards he drew a circle of radius $R$ which center lays inside the rectangle and has coordinates $(x_3, y_3)$. Finally he took a red pencil and outlined the resulting contour.

Help Kostya find the contour's length.

## Input

First line of the input contains four integers $x_1$, $y_1$, $x_2$ and $y_2$ ($-100 \le x_1 < x_2 \le 100$, $-100 \le y_1 < y_2 \le 100$).

Second line contains three integers $x_3$, $y_3$ and $R$ ($x_1 < x_3 < x_2$, $y_1 < y_3 < y_2$, $1 \le R \le 100$).

## Output

Print one float number $P$ — contour's length. The absolute or relative error should not exceed $10^{-6}$.

## Examples

| standard input | standard output |
|---|---|
| 0 0 10 10<br>5 5 6 | 41.5799077631 |
| 0 0 10 10<br>8 9 5 | 48.0160158871 |
| 2 3 6 7<br>4 5 1 | 16.0000000000 |

# Problem D. Increasing sequences

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 64 megabytes |

Recently Little Kostya has learned comparison of integers. The theme was very simple for him, so father told him about new math object — strictly increasing sequences.

The sequence $A = (a_1, a_2, \ldots, a_n)$, $(n > 1)$ is strictly increasing, if the following condition is satisfied $a_1 < a_2 < \ldots < a_n$.

Dad proposed Kostya to practice. He wrote down $M$ pairs of integers $(l_i, r_i)$. For each pair Kostya had to determine whether the sequence $A_i = (a_{l_i}, a_{l_i+1}, \ldots, a_{r_i})$ is strictly increasing.

Kostya spent a lot of time solving the problem. Help him to check it. Find correct answers for this problem.

## Input

First line of the input contains two integers $N$ и $M$ $(1 \le N, M \le 5 \cdot 10^5)$. Second line of the input contains $N$ integers $a_i$ $(1 \le a_i \le 10^6)$. Each of the next $M$ lines contains two integers $l_i$ and $r_i$ $(1 \le l_i \le r_i \le N)$.

## Output

Print one string of $M$ symbols. $i$-th symbol should be `Y`, if the sequence $A_i = (a_{l_i}, a_{l_i+1}, \ldots, a_{r_i})$ is strictly increasing. Otherwise $i$-th symbol should be `N`.

## Examples

| standard input | standard output |
|---|---|
| 7 7<br>1 3 5 6 2 4 7<br>1 7<br>1 4<br>2 5<br>2 4<br>5 7<br>4 7<br>4 4 | NYNYYNN |

# Problem E. Cards

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 64 megabytes |

Little Kostya has $N$ cards with integers from 1 to 1000 (one number per card). Kostya wants to choose some of these cards and arrange them in pairs.

Find the maximal number of pairs of cards Kostya can choose in a such way that for each pair the sum of the numbers written on cards is equal to the same number $S$.

## Input

First line of the input contains one integer number $N$ ($2 \le N \le 500$). Second line of the input contains $N$ integers $a_i$ ($1 \le a_i \le 1000$). Numbers in the line are separated by single spaces.

## Output

In the single line print one integer number — maximal number of pairs of cards with the same sum of number on them.

## Examples

| standard input | standard output |
|---|---|
| 7<br>1 2 3 4 5 6 7 | 3 |
| 10<br>2 6 8 1 3 2 6 6 2 7 | 4 |

# Problem F. Universal numbers

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 5 seconds |
| Memory limit: | 64 megabytes |

Little Kostya likes to play with numbers. Yesterday Dad studied Kostya to calculate integer non-negative powers of numbers. Additionally he asked Kostya to find number of ways to present integer $N$ as sum of three powers with the same base $K$.

In other words, find number of integer non-negative quadruples ($K$, $a$, $b$, $c$) with $K > 0$ such that

$$N = K^a + K^b + K^c.$$

Help Kostya and find this number.

## Input

Only line of the input contains one integer $N$ ($4 \le N \le 10^{18}$).

## Output

Print single integer — required number of quadruples.

## Examples

| standard input | standard output |
|---|---|
| 4 | 3 |
| 5 | 6 |
| 10 | 9 |
| 1234567890 | 4 |

# Problem G. Maximum Product

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 1 second |
| Memory limit: | 256 megabytes |

Find the number from the range $[a, b]$ which has the maximum product of the digits.

## Input

The first line contains two positive integers $a$ and $b$ ($1 \le a \le b \le 10^{18}$) — the left and the right ends of the range. Both of the numbers are giving without trailing spaces.

## Output

Print the number with the maximum product of the digits from the range $[a, b]$.

## Examples

| standard input | standard output |
|---|---|
| 1 10 | 9 |
| 51 62 | 59 |

# Problem H. Biathlon 2.0

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 4 seconds |
| Memory limit: | 256 megabytes |

The Biathlon 2.0 World Championship is held in Petrozavodsk in 2016.

Biathlon 2.0 has different rules, than the classic one. The racing track consists of $n$ segments. The $i$-th segment consists of $a_i$ kilometers to travel and $b_i$ targets to hit. To pass a segment athlete must travel all $a$ kilometers and hit all the targets. The ammo is unlimited and there is no penalty for missing the target. Athletes pass segments in order from 1 to $n$. Each athlete must carry a rifle with him. It is allowed to change rifles between segments.

The national team of Berland has $m$ different rifle types in its disposal. Each rifle is described by two numbers: $c_i$ and $d_i$, where $c_i$ is the number of seconds needed to travel 1 kilometer with this rifle and $d_i$ is the number of seconds needed to hit 1 target with this rifle. The team has unlimited supply of rifles of any type.

Your task is to ensure Berland victory and find a rifle for each track segment to minimize the number of seconds needed to pass the segment.

## Input

The first line contains one integer: $n$ ($1 \le n \le 5 \cdot 10^5$) — the number of segments in the track. The next $n$ lines contain two integers: $a_i, b_i$ ($0 \le a_i, b_i \le 10^9, a_i + b_i > 0$) — the number of kilometers and targets on the $i$-th track segment.

The next line contains one integer: $m$ ($1 \le m \le 5 \cdot 10^5$) — the number of rifle types, used by the national team of Berland.

The next $m$ lines contain two integers: $c_i, d_i$ ($1 \le c_i, d_i \le 10^9$) — the number of seconds needed to travel 1 kilometer with the $i$-th rifle and the number of seconds needed to hit 1 target with the $i$-th rifle.

## Output

Print $n$ space-separated integers. The $i$-th integer is the number of seconds, needed to pass the $i$-th segment if optimal rifle is used.

## Examples

| standard input | standard output |
|---|---|
| 3<br>1 4<br>4 1<br>3 3<br>3<br>1 3<br>3 1<br>2 2 | 7 7 12 |
| 1<br>1000000000 1000000000<br>1<br>1000000000 1000000000 | 2000000000000000000 |

# Problem I. Archaeological Research

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 2 seconds |
| Memory limit: | 256 megabytes |

Professor Tupids has found a mysterious manuscript during his recent archaeological expedition.

The manuscript, in fact, looks like a series of symbols with meaning yet to be disclosed. In order to simplify studying of the manuscript, professor Tupids created an alphabet of all symbols occurred (let's denote their number as $C$) and replaced each symbol with its position in the alphabet (positions are numbered from one). So, the manuscript became represented as a list of $N$ integers from interval $[1; C]$.

Professor's intuition told him that the key for understanding the manuscript is finding some regularities in locations of the symbols. So he wrote down a huge table with $N$ rows and $C$ columns, where cell at $i$-th row and $j$-th column contained a position of the next occurrence of symbol number $j$ after the position $i$ (the cell was left empty if there were no appropriate occurrences).

But then a disaster happened: a fire in the laboratory completely destroyed the manuscript! The table built by professor, luckily, was salvaged — though it was damaged to some extent. Not only some of the cells were lost in the fire, but, thanks to the careless assistants, cells in each of the rows were shuffled randomly!

Professor Tupids doesn't want to lose face in the scientific community, so he asks you to help him with restoring the original manuscript, given the survived information from the table. As there may be infinitely many different solutions (even the size $C$ of the alphabet was lost!), professor wants you to restore the lexicographically smallest solution. It is not guaranteed that the solution exists, though — the table could be completely spoiled by the assistants.

## Input

The first line contains single integer $N$ ($1 \le N \le 3 \cdot 10^5$) — the length of the original manuscript. $N$ lines follow, $i$-th of which contains an integer $c_i$ ($0 \le c_i \le N - i$), followed by $c_i$ distinct integers from interval $[i + 1; N]$ — contents of the survived non-empty cells in the $i$-th row of the table.

It is guaranteed that the sum of all $c_i$ ($1 \le i \le N$) does not exceed $3 \cdot 10^5$.

## Output

If the solution exists, print a single line with $N$ positive integers — the representation of the lexicographically smallest manuscript. Otherwise, print "No solution" (without quotes).

## Examples

| standard input | standard output |
|---|---|
| 4<br>3 2 3 4<br>2 4 3<br>1 4<br>0 | 1 1 2 3 |
| 5<br>1 2<br>1 4<br>1 4<br>1 5<br>0 | 1 1 1 2 1 |

# Problem J. Sockets

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 256 megabytes

Valera has the only electrical socket in his flat. Also he has $m$ devices which require electricity to work. He's got $n$ plug multipliers to plug the devices, the $i$-th plug multiplier has $a_i$ sockets.

The device (or plug multiplier) is plugged to electricity if it is either plugged to the electrical socket, or if it is plugged to some plug multiplier which is plugged to electricity.

For every device Valera knows the safety value $b_j$ — the maximal length of plug multipliers chain to connect the $j$-th device to electricity. For example, if $b_j = 0$ the device should be directly plugged into the socket in his flat.

What is the maximum number of devices Valera could plug to electricity? Note that all devices and plug multipliers take one socket to plug and that he can use a socket to plug either one device or one plug multiplier.

## Input

The first line contains two space-separated integers $n$ and $m$ ($1 \le n, m \le 2 \cdot 10^5$) — the number of plug multipliers and the number of devices correspondingly.

The second line contains $n$ space-separated integers $a_1, a_2, \ldots, a_n$ ($2 \le a_i \le 2 \cdot 10^5$) — number $a_i$ stands for the number of sockets on the $i$-th plug multiplier.

The third line contains $m$ space-separated integers $b_1, b_2, \ldots, b_m$ ($0 \le b_j \le 2 \cdot 10^5$) — number $b_j$ stands for the safety value of the $j$-th device.

## Output

Print a single integer — the maximum number of devices that could be plugged to electricity.

## Examples

| standard input | standard output |
|---|---|
| 3 5<br>3 2 2<br>1 2 2 1 1 | 4 |
| 3 3<br>2 2 2<br>1 2 2 | 3 |

# Problem K. Toll Roads

| | |
|---|---|
| Input file: | `standard input` |
| Output file: | `standard output` |
| Time limit: | 10 seconds |
| Memory limit: | 256 megabytes |

There are $n$ cities in the country of Flatland, $n-1$ bidirectional roads connect some pairs of Flatland's cities. It is possible to get from any city to any other city of the country using roads. In this problem we will refer to the smallest set of roads needed to travel from $a$ to $b$ as a *simple path*.

Government of Flatland has recently introduced payment tolls on all roads. Using one road connecting two cities costs 1 ruble. That means that every time when you travel from one city to another one, you have to pay as many rubles, as the number of roads on your journey.

Many people got upset because of this government's decision, especially people who travel on long distances. Political opponents of the current government claim that the maximal cost of a simple path in Flatland is very high, namely *cost* rubles. Government has decided to support people and calm down the opposition. They want to reduce the maximal cost of a simple path in the country as much as possible. In other words, they want the number *cost* reported by the opposition to be lowest possible. Government will allow to remove tolls from at most $k$ roads in order to achieve that. If there are multiple ways to do that, they want minimize the number of roads that will become free.

Like any other government, Flatland's one is quite inflexible. They additionally require that the set of roads which will become free can be represented as a simple path between some cities $x$ and $y$. Your task is to help the government.

## Input

The first line contains two integers $n$ and $k$ ($1 \le k < n \le 5000$) — number of cities in Flatland and maximum number of roads to become toll-free, respectively. Next $n-1$ lines contain two integers each: $u_i$, $v_i$, meaning that there is a road connecting $u_i$ and $v_i$ ($0 \le u_i, v_i < n$).

## Output

On the first line of the output print one integer: minimal *cost* of the most expensive simple path in Flatland that the government can achieve ($0 \le cost < n-1$). On the second line print the minimal number of roads $t$ that must be made free to achieve that *cost* ($0 \le t \le k$). If $t > 0$, print two space-separated numbers on the third line: $x$ and $y$. These cities specify which $t$ roads must be made free ($0 \le x, y < n$, simple path between $x$ and $y$ must contain $k$ roads).

## Examples

| standard input | standard output |
|---|---|
| 8 3<br>0 2<br>0 5<br>2 3<br>5 1<br>4 5<br>5 6<br>6 7 | 2<br>3<br>2 6 |
| 5 2<br>0 1<br>0 2<br>0 3<br>0 4 | 2<br>0 |