icpc | asia pacific contests

The 2023 ICPC
Asia Yokohama Regional Contest

Official Problem Set

asia pacific

2023 2024

icpc.foundation

JET BRAINS

icpc global sponsor
programming tools

HUAWEI

icpc diamond
multi-regional sponsor

Problem A
Yokohama Phenomena
Time Limit: 2 seconds

Do you know about Yokohama Phenomena? The phenomenon takes place when three programmers, sitting around a table, hold a single pen together above a board. A grid of squares is drawn on the board, with each square marked with a single letter. Although none of the participants purposely moves the pen, its nib, as if it has a will, goes down to one of the squares marked with `Y`, and then starts moving on the board. The squares passed are marked with `O`, `K`, `O`, `H`, `A`, and `M` in this order, and then the nib stops on the square marked with `A`.

Let us call the series of squares along such a trajectory of the nib a *YOKOHAMA trace*. A YOKOHAMA trace is defined as follows.

- It is a series of eight squares in the given grid of squares.
- Every square in the series, except for the first one, shares an edge with (is edge-adjacent to) its directly preceding square in the series.
- The letters marked in the eight squares of the series are `Y`, `O`, `K`, `O`, `H`, `A`, `M`, and `A`, in this order.

Note that the same square may appear more than once in the series.

Figure A.1 (a) is an illustration of the board corresponding to Sample Input 1. Figures A.1 (b) and (c) show trajectories on two of the YOKOHAMA traces. Both traces start at the leftmost square in the upper row. The same square marked with `O` appears twice in the trace illustrated in Figure A.1 (c).

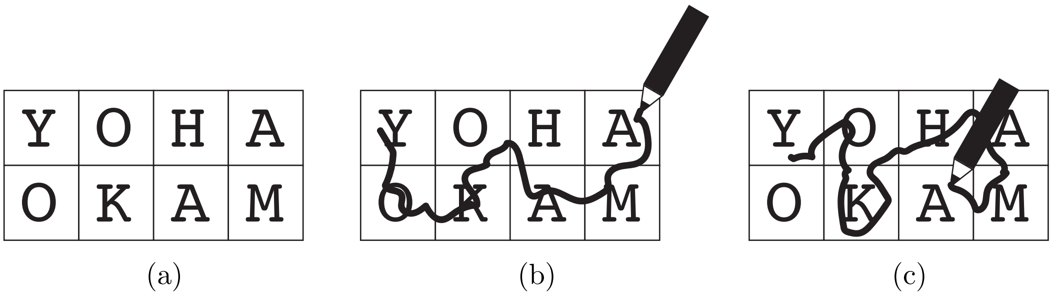

Figure A.1. A board and trajectories on two of the YOKOHAMA traces

You are given a grid of squares, each marked with one of six letters, `A`, `H`, `K`, `M`, `O`, or `Y`. Your task is to count how many distinct YOKOHAMA traces are possible on it.

Input

The input consists of a single test case of the following format.

$$
\begin{array}{l}
n\ m \\
x_{1,1}\ \cdots\ x_{1,m} \\
\vdots \\
x_{n,1}\ \cdots\ x_{n,m}
\end{array}
$$

The first two integers n and m ($1 \leq n \leq 10$, $1 \leq m \leq 10$) describe the size of the grid. The grid has squares arranged in an $n \times m$ matrix. The following n lines describe the letters marked in the squares. The square at the i-th row and the j-th column in the grid ($1 \leq i \leq n$, $1 \leq j \leq m$) has letter $x_{i,j}$ marked in it. Each $x_{i,j}$ is one of the six letters, `A`, `H`, `K`, `M`, `O`, or `Y`.

Output

Output a line containing the number of distinct YOKOHAMA traces.

Sample Input 1	Sample Output 1
2 4 YOHA OKAM	8

Sample Input 2	Sample Output 2
3 4 YOKH OKHA KHAM	0

Sample Input 3	Sample Output 3
3 6 MAYOHA AHOKAM MAYOHA	80

Problem B
Rank Promotion
Time Limit: 3 seconds

Quiz Solver is a popular online computer game. Each time a player opens the mobile application of the game, a new quiz is displayed. The player submits an answer to the quiz, and then it is judged either as correct or incorrect, which is accumulated in the database. When the player shows high accuracy for a number of quizzes, the *rank* of the player in the game is promoted.

Player ranks are non-negative integers, and each player starts the game at the rank 0. The player will be promoted to the next rank when the player achieves a high ratio of correct answers during a sufficiently long sequence of quizzes. More precisely, the rank promotion system is defined by two parameters: an integer c, and a rational number p/q. After finishing the e-th quiz, the player's rank is immediately incremented by one if there exists an integer s satisfying the following conditions.

- $1 \leq s \leq e - c + 1$.
- The player was already at the current rank before starting the s-th quiz.
- The ratio of correct answers of the quizzes from the s-th through the e-th is higher than or equal to p/q.

Otherwise, the rank stays the same.

One day, the administrator of Quiz Solver realized that the rank data of the players were lost due to a database failure. Luckily, the log of quiz solving records was completely secured without any damages. Your task is to recompute the rank of each player from the solving records for the player.

Input

The input consists of a single test case in the following format.

$n\ c\ p\ q$
$S_1 \cdots S_n$

The first line consists of four integers satisfying the following constraints: $1 \leq n \leq 5 \times 10^5$, $1 \leq c \leq 200$, and $1 \leq p \leq q \leq 5 \times 10^5$. The first integer n is the number of quizzes answered by a single player. The meanings of the parameters c, p, and q are described in the problem statement.

$S_1 \cdots S_n$ is a string describing the quiz solving records of the player. Each S_i is either `Y` meaning that the player's answer for the i-th quiz was correct, or `N` meaning incorrect.

Output

Output the final rank of the player after finishing the n-th quiz in one line.

Sample Input 1	Sample Output 1
`12 4 4 7` `YYYNYYNNNYYN`	`2`

Sample Input 2	Sample Output 2
`10 1 1 1` `YNYNYNYNYN`	`5`

Sample Input 3	Sample Output 3
`17 5 250000 500000` `YYYYYYYYYYYYYYYYY`	`3`

Sample Input 4	Sample Output 4
`8 3 2 3` `YNNYYYYN`	`2`

In Sample Input 1, the player is promoted to the rank 1 after finishing the fourth quiz, because the ratio of the correct answers $3/4$ is higher than $p/q = 4/7$. Note that, the promotion didn't happen at the third quiz because only three quizzes had been answered, which is less than $c = 4$. Then, after the eleventh quiz, the player is promoted to the rank 2.

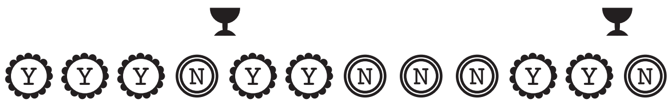

Figure B.1. The timings of rank promotions of Sample Input 1

Problem C
Ferris Wheel
Time Limit: 6 seconds

The big Ferris wheel, *Cosmo Clock 21*, is a landmark of Yokohama and adds beauty to the city's night view. The ICPC city also wants something similar.

The ICPC city plans to build an illuminated Ferris wheel with an even number of gondolas. All the gondolas are to be colored with one of the given set of candidate colors. The illumination is planned as follows.

- All the gondolas are paired up; every gondola belongs to a single pair.
- Only two gondolas of the same color can form a pair.
- Paired gondolas are connected with a straight LED line to illuminate the wheel.
- No two LED lines cross when looked from the front side.

A coloring of gondolas is *suitable* if it allows at least one way of pairing for the illumination plan.

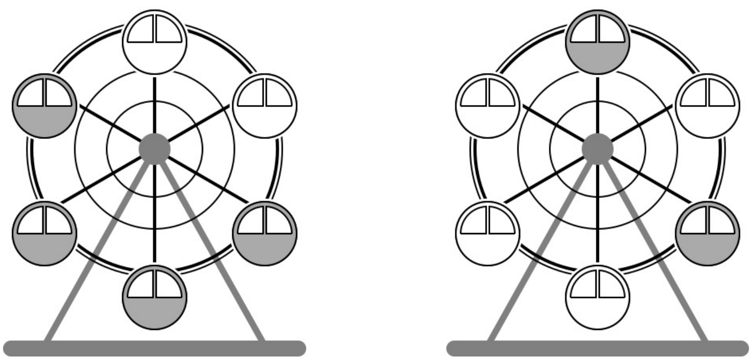

Figure C.1. Ferris wheels with suitable (left) and *not* suitable (right) colorings

Given the numbers of gondolas and candidate colors, count the number of suitable colorings of gondolas. Since the Ferris wheel rotates, two colorings are considered the same if they coincide under a certain rotation. Two colorings that coincide only when looked from the opposite sides are considered different.

Input

The input consists of a single test case in the following format.

n k

n and k are integers between 1 and 3×10^6, inclusive. The numbers of gondolas and candidate colors are $2n$ and k, respectively.

Output

Output the number of suitable colorings of gondolas in modulo $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$, which is a prime number.

Sample Input 1	Sample Output 1
`3 2`	`6`

Sample Input 2	Sample Output 2
`5 3`	`372`

Sample Input 3	Sample Output 3
`2023 1126`	`900119621`

For Sample Input 1, there are six suitable colorings as listed in the figure below.

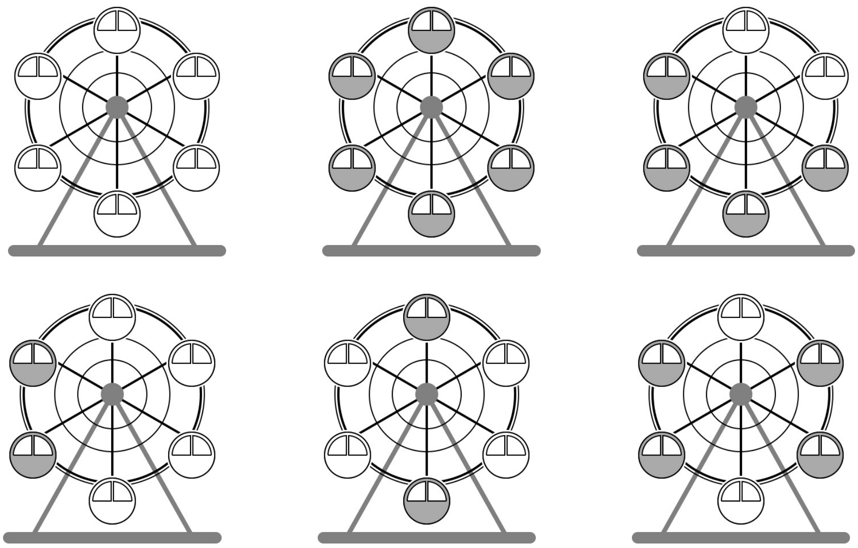

Figure C.2. Suitable colorings in case of $n = 3$ and $k = 2$

Problem D
Nested Repetition Compression
Time Limit: 2 seconds

You have a number of strings of lowercase letters to be sent in e-mails, but some of them are so long that typing as they are would be tiresome. As you found repeated parts in them, you have decided to try out a simple compression method in which repeated sequences are enclosed in parentheses, prefixed with digits meaning the numbers of repetitions. For example, the string "`abababaaaaa`" can be represented as "`3(ab)5(a)`" or "`a3(ba)4(a)`". The syntax of compressed representations is given below in Backus-Naur form with the start symbol S.

<*S*> ::= <*R*> | <*R*><*S*>
<*R*> ::= <*L*> | <*D*> `(` <*S*> `)`
<*D*> ::= `2` | `3` | `4` | `5` | `6` | `7` | `8` | `9`
<*L*> ::= `a` | `b` | `c` | `d` | `e` | `f` | `g` | `h` | `i` | `j` | `k` | `l` | `m` | `n` | `o` | `p` | `q` | `r` | `s` | `t` | `u` | `v` | `w` | `x` | `y` | `z`

Note that numbers of repetitions are specified by a single digit, and thus at most nine, but more repetitions can be specified by nesting them. A string of thirty `a`'s can be represented as "`6(5(a))`" or "`3(5(2(a)))`", for example.

Find the shortest possible representation of the given string in this compression scheme.

Input

The input is a line containing a string of lowercase letters. The number of letters in the string is at least one and at most 200.

Output

Output a single line containing the shortest possible representation of the input string. If there exist two or more shortest representations, any of them is acceptable.

Sample Input 1	Sample Output 1
`abababaaaaa`	`3(ab)5(a)`

Sample Input 2	Sample Output 2
`abababcaaaaaabababcaaaaa`	`2(3(ab)c5(a))`

Sample Input 3	Sample Output 3
`abcdefg`	`abcdefg`

Problem E
Chayas
Time Limit: 8 seconds

Once upon a time, there were a number of *chayas* (teahouses) along one side of an east-west road in Yokohama. Although the total number of chayas is known, the information about their locations was considered to be lost totally.

Recently, a document describing the old townscapes of Yokohama has been found. The document contains a number of records on the order of the locations of chayas. Each record has information below on the order of the locations of three chayas, say a, b, and c.

> Chaya b was located between chayas a and c. Note that there may have been other chayas between a and b, or between b and c. Also, note that chaya a may have been located east of c or west of c.

We want to know how many different orders of chayas along the road are consistent with all of these records in the recently found document. Note that, as the records may have some errors, there might exist no orders consistent with the records.

Input

The input consists of a single test case given in the following format.

n m
a_1 b_1 c_1
$\vdots$
a_m b_m c_m

Here, n represents the number of chayas and m represents the number of records in the recently found document. $3 \le n \le 24$ and $1 \le m \le n \times (n-1) \times (n-2)/2$ hold. The chayas are numbered from 1 to n.

Each of the following m lines represents a record. The i-th of them contains three distinct integers a_i, b_i, and c_i, each between 1 and n, inclusive. This says that chaya b_i was located between chayas a_i and c_i. No two records have the same information, that is, for any two different integers i and j, the triple (a_i, b_i, c_i) is not equal to (a_j, b_j, c_j) nor (c_j, b_j, a_j).

Output

Output the number of different orders of the chayas, from east to west, consistent with all of the records modulo 998 244 353 in a line. Note that $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$ is a prime number.

Sample Input 1	Sample Output 1

`5 4`
`1 2 4`
`2 3 5`
`3 2 4`
`1 3 2`
	`4`

Sample Input 2	Sample Output 2

`4 2`
`3 1 4`
`1 4 3`
	`0`

For Sample Input 1, four orders, $(1, 5, 3, 2, 4)$, $(4, 2, 3, 1, 5)$, $(4, 2, 3, 5, 1)$, and $(5, 1, 3, 2, 4)$, are consistent with the records.

For Sample Input 2, there are no consistent orders.

Problem F
Color Inversion on a Huge Chessboard
Time Limit: 4 seconds

You are given a set of square cells arranged in a chessboard-like pattern with n horizontal rows and n vertical columns. Rows are numbered 1 through n from top to bottom, and columns are also numbered 1 through n from left to right.

Initially, the cells are colored as in a chessboard, that is, the cell in the row i and the column j is colored black if $i + j$ is odd and is colored white if it is even.

Color-inversion operations, each of which is one of the following two, are made one after another.

Invert colors of a row: Given a row number, invert colors of all the cells in the specified row. The white cells in the row become black and the black ones become white.

Invert colors of a column: Given a column number, invert colors of all the cells in the specified column. The white cells in the column become black and the black ones become white.

The number of distinct *areas* after each of the operations should be counted. Here, an area means a group of directly or indirectly connected cells of the same color. Two cells are said to be directly connected when they share an edge.

Input

The input consists of a single test case of the following format.

n q
$operation_1$
$\vdots$
$operation_q$

The integer n is the number of rows and columns ($1 \le n \le 5 \times 10^5$). The integer q is the number of operations ($1 \le q \le 5 \times 10^5$). The following q lines represent operations to be made in this order. Each of them is given in either of the following forms.

- `ROW` i: the operation "invert colors of a row" applied to the row i ($1 \le i \le n$).
- `COLUMN` j: the operation "invert colors of a column" applied to the column j ($1 \le j \le n$).

Output

Output q lines. The k-th line should contain an integer denoting the number of areas after the k-th operation is made.

Sample Input 1

```
3 3
ROW 2
COLUMN 3
ROW 2
```

Sample Output 1

```
3
2
6
```

Sample Input 2

```
200000 2
ROW 1
ROW 1
```

Sample Output 2

```
39999800000
40000000000
```

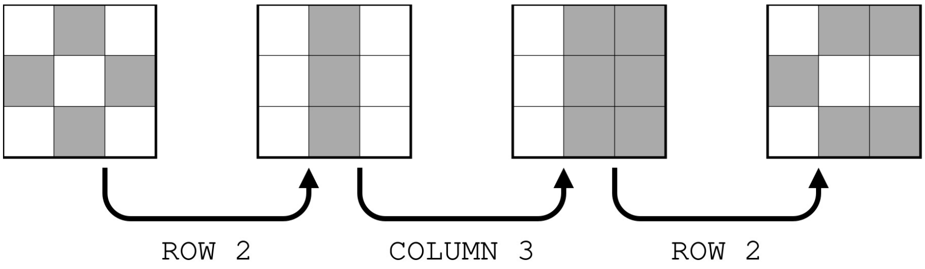

Figure F.1. Illustration of Sample Input 1

Problem G
Fortune Telling
Time Limit: 5 seconds

Your fortune is to be told by a famous fortune teller. She has a number of tarot cards and a six-sided die. Using the die, she will choose one card as follows and that card shall tell your future.

Initially, the cards are lined up in a row from left to right. The die is thrown showing up one of the numbers from one through six with equal probability. When x is the number the die shows up, the x-th card from the left and every sixth card following it, i.e., the $(x + 6k)$-th cards for $k = 0, 1, 2, \ldots$, are removed and then remaining cards are slid left to eliminate the gaps. Note that if the number of cards remaining is less than x, no cards are removed. This removing and sliding procedure is repeated until only one card remains.

Figure G.1 illustrates how cards are removed and slid when the die shows up two.

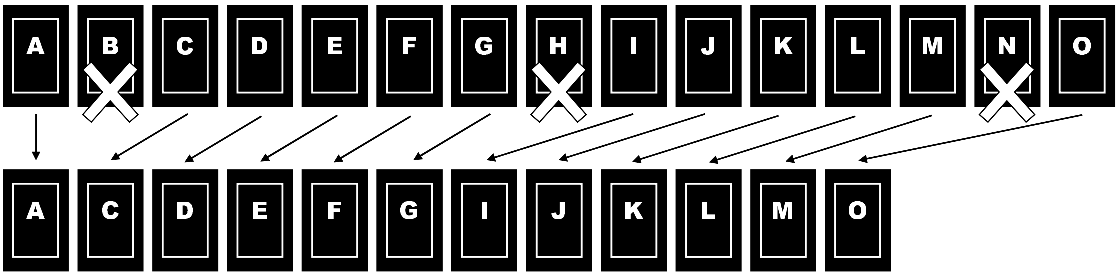

Figure G.1. Removing and sliding cards

You are given the number of initial tarot cards. For each card initially placed, compute the probability that the card will remain in the end.

Input

The input is a single line containing an integer n, indicating the number of tarot cards, which is between 2 and 3×10^5, inclusive.

Output

Output n lines, the i-th of which should be an integer that is determined, as follows, by the probability of the i-th card from the left to remain in the end.

It can be proved that the probability is represented as an irreducible fraction a/b, where b is not divisible by a prime number $998\,244\,353 = 2^{23} \times 7 \times 17 + 1$. There exists a unique integer c such that $bc \equiv a \pmod{998\,244\,353}$ and $0 \le c < 998\,244\,353$. What should be output is this integer c.

Sample Input 1	Sample Output 1
3	332748118 332748118 332748118

Sample Input 2	Sample Output 2
7	305019108 876236710 876236710 876236710 876236710 876236710 305019108

Sample Input 3	Sample Output 3
8	64701023 112764640 160828257 160828257 160828257 160828257 112764640 64701023

For Sample Input 1, the probabilities to remain in the end for all the cards are equal, that are $1/3$.

For Sample Input 2, let us consider the probability of the leftmost card to remain in the end. To make this happen, the first number the die shows up should not be one. After getting a number other than one, six cards will remain. Each of these six cards will remain in the end with the same probability. From this observation, the probability of the leftmost card to remain in the end is computed as $(5/6) \times (1/6) = 5/36$. The same argument holds for the rightmost card. As for the rest of the cards, the probabilities are equal, and they are $(1 - 2 \times 5/36)/5 = 13/90$.

Problem H
Task Assignment to Two Employees
Time Limit: 2 seconds

Hanako is the CEO of a small company with two employees. She currently has some number of tasks and aims to earn some profits by making the employees do the tasks. Employees can enhance their skills through the tasks and, with higher skills, a larger profit can be earned from the same task. Thus, assigning tasks to appropriate employees in an appropriate order is important for maximizing the total profit.

For each pair (i, j) of employee i and task j, two non-negative integers $v_{i,j}$ and $s_{i,j}$ are defined. Here, $v_{i,j}$ is the task compatibility and $s_{i,j}$ is the amount of skill growth. When task j has been completed by employee i whose skill point was p, a profit of $p \times v_{i,j}$ is earned, and his skill point increases to $p + s_{i,j}$. Initially, both employees have skill points of p_0.

Note that the skill points are individual, and completing a task by one employee does not change the skill point of the other. Each task must be done only once by only one employee. The order of tasks to carry out can be arbitrarily chosen.

Input

The input consists of a single test case of the following format.

$$
\begin{array}{l}
n \ p_0 \\
s_{1,1} \ \cdots \ s_{1,n} \\
s_{2,1} \ \cdots \ s_{2,n} \\
v_{1,1} \ \cdots \ v_{1,n} \\
v_{2,1} \ \cdots \ v_{2,n}
\end{array}
$$

All the input items are non-negative integers. The number of tasks n satisfies $1 \leq n \leq 100$. The initial skill point p_0 satisfies $0 \leq p_0 \leq 10^8$. Each $s_{i,j}$ is the amount of skill growth for the employee i by completing the task j, which satisfies $0 \leq s_{i,j} \leq 10^6$. Each $v_{i,j}$ is the task compatibility of the employee i with the task j, which satisfies $0 \leq v_{i,j} \leq 10^6$.

Output

Output the maximum possible total profit in one line.

Sample Input 1	**Sample Output 1**

`4 0`
`10000 1 1 1`
`1 1 10000 1`
`1 10000 1 1`
`1 1 1 10000`
	`200000000`

Sample Input 2	**Sample Output 2**

`3 1`
`1 1 1`
`2 2 2`
`2 2 2`
`1 1 1`
	`12`

Problem I
Liquid Distribution
Time Limit: 2 seconds

After years of space exploration, humans succeeded in bringing back a small amount of sample materials from an asteroid to Earth! The materials were stored in several bottles, each containing a complete mixture of two liquids, A and B.

Intense discussion finally reached an agreement that all the materials brought back should be distributed to the research institutes participated in the exploration. The amounts of the liquids A and B to be sent were decided depending on the research topics of each of the institutes.

However, after this decision, a problem was found that it is impossible with current human technologies to separate two liquids from the mixture. The only operations we can perform are to take some amounts of the mixtures from one or more bottles and put them together in a new bottle.

Your task is to judge whether the agreed distribution of the liquids is possible ever.

Input

The input consists of a single test case of the following format.

n m
$a_1 \cdots a_n$
$b_1 \cdots b_n$
$c_1 \cdots c_m$
$d_1 \cdots d_m$

Here, n is the number of the bottles initially containing the mixtures of liquid A and liquid B, while m is the number of the research institutes to which liquids are to be sent. Both n and m are positive integers not greater than 500. The following two lines contain n integers each, meaning that the i-th bottle ($1 \leq i \leq n$) initially contains the mixture of a_i mL of liquid A and b_i mL of liquid B. The following two lines contain m integers each, meaning that a bottle containing c_j mL of liquid A and d_j mL of liquid B is to be sent to the j-th institute ($1 \leq j \leq m$). All of a_i, b_i, c_j and d_j are positive integers not greater than 10^6. Both $\sum_{i=1}^{n} a_i = \sum_{j=1}^{m} c_j$ and $\sum_{i=1}^{n} b_i = \sum_{j=1}^{m} d_j$ hold.

Output

If the agreed distribution is possible, output `Yes`; otherwise, output `No` in a line.

Sample Input 1	**Sample Output 1**
2 2 1 3 3 1 2 2 1 3	Yes

Sample Input 2	**Sample Output 2**
2 2 2 2 2 2 1 3 3 1	No

Sample Input 3	**Sample Output 3**
3 5 2 5 8 3 5 7 3 3 3 3 3 3 3 3 3 3	Yes

Sample Input 4	**Sample Output 4**
3 2 4 4 4 1 9 5 6 6 2 13	No

For Sample Input 1, the only way that conforms to the decision is to send 0.5 mL from the bottle 1 and 2.5 mL from the bottle 2 put together in a bottle to the institute 1, and a bottle of the rest to the institute 2.

For Sample Input 2, the distribution agreement cannot be fulfilled.

Problem J
Do It Yourself?
Time Limit: 10 seconds

You are the head of a group of n employees including you in a company. Each employee has an employee ID, which is an integer 1 through n, where your ID is 1. Employees are often called by their IDs for short: #1, #2, and so on. Every employee other than you has a unique *immediate boss,* whose ID is smaller than his/hers. A *boss* of an employee is transitively defined as follows.

- If an employee $\#i$ is the immediate boss of an employee $\#j$, then $\#i$ is a boss of $\#j$.
- If $\#i$ is a boss of $\#j$ and $\#j$ is a boss of $\#k$, then $\#i$ is a boss of $\#k$.

Every employee including you is initially assigned exactly one task. That task can be done by him/herself or by any one of his/her bosses. Each employee can do an arbitrary number of tasks, but doing many tasks makes the employee too tired. Formally, each employee $\#i$ has an individual fatigability constant f_i, and if $\#i$ does m tasks in total, then the *fatigue level* of $\#i$ will become $f_i \times m^2$.

Your mission is to minimize the sum of fatigue levels of all the n employees.

Input

The input consists of a single test case in the following format.

$$
\begin{array}{l}
n \\
b_2\ b_3\ \cdots\ b_n \\
f_1\ f_2\ \cdots\ f_n
\end{array}
$$

The integer n in the first line is the number of employees, where $2 \le n \le 5 \times 10^5$. The second line has $n-1$ integers b_i $(2 \le i \le n)$, each of which represents the immediate boss of the employee $\#i$, where $1 \le b_i < i$. The third line has n integers f_i $(1 \le i \le n)$, each of which is the fatigability constant of the employee $\#i$, where $1 \le f_i \le 10^{12}$.

Output

Output the minimum possible sum of fatigue levels of all the n employees.

Sample Input 1	Sample Output 1
4 1 1 2 1 1 1 1	4

Sample Input 2	Sample Output 2
4 1 1 2 1 10 10 10	16

Sample Input 3	Sample Output 3
4 1 1 2 1 2 4 8	10

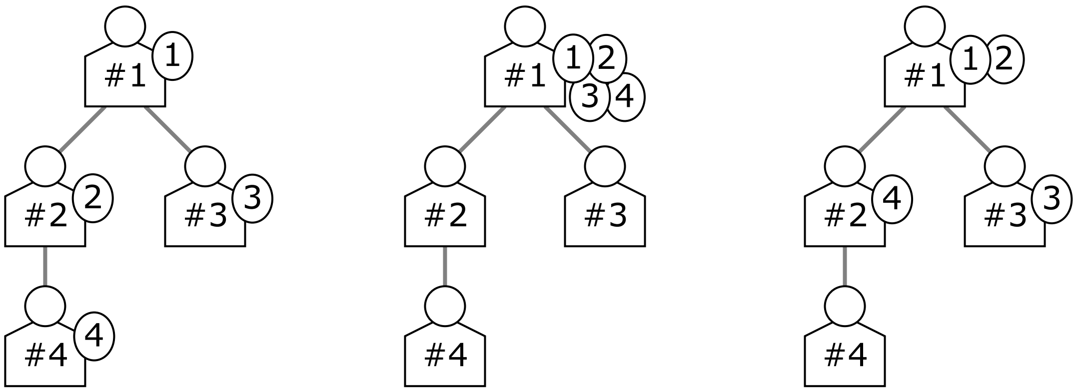

Figure J.1. Illustration of the three samples (from left to right)

The situations and solutions of the three samples are illustrated in Figure J.1.

For Sample Input 1, the unique optimal way is that each employee does his/her task by him/herself. That is, you should just say "Do it yourself!" to everyone.

For Sample Input 2, the unique optimal way is that the employee #1 does all the tasks. That is, you should just say "Leave it to me!" to everyone.

For Sample Input 3, one of the optimal ways is the following.

- #1 does the tasks of #1 and #2, and then the fatigue level of #1 will be $1 \times 2^2 = 4$.
- #2 does the task of #4, and then the fatigue level of #2 will be $2 \times 1^2 = 2$.
- #3 does the initially assigned task, and then the fatigue level of #3 will be $4 \times 1^2 = 4$.
- #4 does nothing, and then the fatigue level of #4 will be $8 \times 0^2 = 0$.

Thus, the sum of the fatigue levels is $4 + 2 + 4 + 0 = 10$. There is another optimal way, in which the employees #1, #2, and #3 do their initially assigned tasks by themselves and #1 does the task of #4 in addition.

Problem K
Probing the Disk
Time Limit: 2 seconds

This is an interactive problem.

A thin black disk is laid flat on the square bottom of a white box. The sides of the box bottom are 10^5 units long.

Somehow, you are not allowed to look into the box, but you want to know how large the disk is and where in the box bottom the disk is laid. You know that the shape of the disk is a true circle with an integer units of radius, not less than 100 units, and its center is integer units distant from the sides of the box bottom. The radius of the disk is, of course, not greater than the distances of the center of the disk from any of the sides of the box bottom.

You can probe the disk by projecting a thin line segment of light to the box bottom. As the reflection coefficients of the disk and the box bottom are quite different, from the overall reflection intensity, you can tell the length of the part of the segment that lit the disk.

Your task is to decide the exact position and size of the disk through repetitive probes.

Interaction

You can repeat probes, each of which is a pair of sending a query and receiving the response to it. You can probe at most 1024 times.

A query should be sent to the standard output in the following format, followed by a newline.

`query` x_1 y_1 x_2 y_2

Here, (x_1, y_1) and (x_2, y_2) are the positions of the two ends of the line segment of the light. They have to indicate distinct points. The coordinate system is such that one of the corners of the box bottom is the origin $(0, 0)$ and the diagonal corner has the coordinates $(10^5, 10^5)$. All of x_1, y_1, x_2, and y_2 should be integers between 0 and 10^5, inclusive.

In response to this query, a real number is sent back to the standard input, followed by a newline. The number indicates the length of the part of the segment that lit the disk. It is in decimal notation without exponent part, with 7 digits after the decimal point. The number may contain an absolute error up to 10^{-6}.

When you become sure about the position and the size of the disk through the probes, you can send your answer. The answer should have the center position and the radius of the disk. It should be sent to the standard output in the following format, followed by a newline.

`answer` x y r

Here, (x, y) should be the position of the center of the disk, and r the radius of the disk. All of x, y, and r should be integers.

After sending the answer, your program should terminate without any extra output. Thus, you can send the answer only once.

Notes on interactive judging

When your output violates any of the conditions above (incorrect answer, invalid format, x_1, y_1, x_2, or y_2 being out of the range, too many queries, any extra output after sending your answer, and so on), your submission will be judged as a wrong answer. As some environments require flushing the output buffers, make sure that your outputs are actually sent. Otherwise, your outputs will never reach the judge.

You are provided with a command-line tool for local testing. For more details, refer to the clarification in the contest system.

Sample Interaction

Read	Write
	`query 40000 0 40000 100000`
`60000.0000000`	
	`query 0 10000 100000 10000`
`0.0000000`	
	`query 60000 60000 80000 80000`
`12315.3774869`	
	`answer 40000 60000 30000`

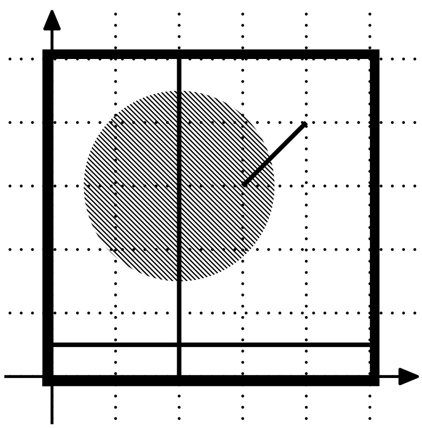

Figure K.1. Sample Interaction