Problem A. Attack and Defence

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	128 mebibytes

In Tower Defence games, you should build some towers to protect your kingdom from monsters. And now another wave of monsters is coming and you need again to know whether you can get through it.

The path of monsters is a straight line, and there are N blocks on it (numbered from 1 to N continuously). Before enemies come, you have M towers built. Each tower has an attack range $[L, R]$, meaning that it can attack all enemies in every block i, where $L \leq i \leq R$. Once a monster steps into block i, every tower whose attack range include block i will attack the monster once and only once. For example, a tower with attack range $[1, 3]$ will attack a monster three times if the monster is alive, one in block 1, another in block 2 and the last in block 3.

A witch helps your enemies and makes every monster has its own place of appearance (the i-th monster appears at block X_i). All monsters go straightly to block N.

Now that you know each monster has HP H_i and each tower has a value of attack D_i, one attack will cause D_i damage (decrease HP by D_i). If the HP of a monster is decreased to 0 or below 0, it will die and disappear.

Your task is to calculate the number of monsters surviving from your towers so as to make a plan B.

Input

The first line of the input is an integer N ($0 < N <\leq 10^5$), the number of blocks in the path. The second line is an integer M ($0 < M \leq 10^5$), the number of towers you have. The next M lines each contain three numbers, L_i, R_i, D_i ($1 \leq L_i \leq R_i \leq N$, $0 < D_i \leq 1000$), indicating the attack range $[L, R]$ and the value of attack D of the i-th tower. The next line is an integer K ($0 < K \leq 10^5$), the number of coming monsters. The following K lines each contain two integers H_i and X_i ($0 < H_i \leq 10^{18}$, $1 \leq X_i \leq N$) indicating the i-th monster's live point and the number of the block where the i-th monster appears.

Output

Output one line containing the number of surviving monsters.

Example

standard input	standard output

5
2
1 3 1
5 5 2
5
1 3
3 1
5 2
7 3
9 1
	3

Problem B. Build The Dice

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Lets call a dice *correct*, if sum of scores on each pair of parallel faces is the same. For example, dice with 0 on the upper face, 3 on the lower face, 2 on the right face, 1 on the left face, 1 on the back face, 2 on the front face is correct, while dice with 1 on the upper face, 6 on the lower face, 2 on the right face, 5 on the left face, 4 on the back face, 7 on the front face is incorrect (sums are 7, 7 and 11). You are given six non-negative integers. Check if you can find a way to put all those integers on faces of the cube to form correct dice.

Input

The first and only line of the input file contains six non-negative integers, each of them is not greater than 1000.

Output

Print "`Yes`", if it is possible to form the correct dice with those six numbers on the faces, or "`No`" otherwise.

Examples

standard input	standard output
`0 1 3 2 2 1`	`Yes`
`1 2 4 5 6 7`	`No`

Problem C. Cloud Computing

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Nowadays, cloud computing is a popular topic among business and technology circles. In cloud computing, everything can be stored in a cloud server, and clients deal with the data by sending operation-requests to the server. However, such client-server mode may cause some problems. For example, if you send two operations $op1$ and $op2$ to the server, expecting that $op1$ should be executed first and followed by $op2$. Due to the network delay, $op1$ may arrive at the server later than $op2$. In order to inform the server the correct operation order, each operation is associated with a timestamp. Now if the server gets two operations $(op1, t_1)$ and $(op2, t_2)$, where $t_1 < t_2$, the server will know that $op1$ should be executed earlier than $op2$. So, if $(op2, t_2)$ arrives first, the server will execute $op2$ immediately. And when $(op1, t_1)$ arrives, the server will find that $op1$ should be executed before $op2$ (because $t_1 < t_2$), thus it has to undo $op2$, execute $op1$, and re-execute $op2$ finally.

In this problem, you are asked to simulate the above process. To simplify the problem, we assume that there is only a stack, a last-in-first-out data structure as you know, stored in the server. Three types of operations are considered, whose formats and semantics are given as follows.

- `push x t` – push x into the stack, and t is a timestamp;
- `pop t` – pop the top element from the stack, and t is a timestamp;
- `peak t` – return the top element in the stack, and t is a timestamp;

When an operation op with a timestamp t arrives, the server process it in the following three steps:

Step 1: undo all the "`push`" and "`pop`" operations having timestamp larger than t.

Step 2: execute op.

Step 3: redo all the "`push`" and "`pop`" operations which were undone in step 1.

The server do not need to undo or redo any "`peak`" operations. In another word, every "`peak`" operation is executed only once after it arrives at the server.

Given the operations arriving at the server in order, you are asked to simulate the above process. The stack is empty initially. To simplify the problem further, another two assumptions are made:

1. All the "`pop`" operations are valid. In another word, if you simulate the process correctly, no "`pop`" operations will be performed on an empty stack.
2. All timestamps are different.

Input

The input begins with an integer N ($1 \le N \le 5 \cdot 10^4$), indicating the number of operations. The following N lines each contain an operation in one of the following three formats: "`push x t`", "`pop t`", "`peak t`", where $0 \le x, t \le 10^9$.

The operations are given in the order in which they arrive at the server.

Output

For each "`peak`" operation, output the answer in a line. If the stack is empty, output -1 instead.

Examples

standard input	standard output
7	100
push 100 3	50
push 200 7	200
peak 4	
push 50 2	
pop 5	
peak 6	
peak 8	
4	-1
push 25 1	25
pop 5	
peak 6	
peak 3	
4	10
push 10 1	-1
peak 7	
pop 3	
peak 4	

Problem D. Divide The String

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Let S be a string of length $2N$, where each character in S is assigned an integer as its weight. The weight of a subsequence T of S (denoted by $weight(T)$), is defined as the sum of all its characters' weights. Your task is to divide S into two subsequences T_1 and T_2, each of length N, such that T_1 is equal to T_2 and $|weight(T_1) - weight(T_2)|$ is as small as possible.

Input

The first line of the input contains an integer N ($1 \le N \le 20$). The second line is a string S of length $2N$. Each character in S is either '`a`' or '`b`'. The third line contains $2N$ positive integers w_i ($1 \le w_i \le 10^6$), where the i-th integer is the weight of the ith character in S.

Output

Print the smallest difference between $weight(T_1)$ and $weight(T_2)$ in a line. If it is impossible to divide S into two equal subsequences, output -1 instead.

Examples

standard input	standard output
2 abab 3 1 10 5	11
3 aaabbb 1 1 1 2 2 2	-1
3 abaaba 4 6 10 5 3 4	2

Problem E. Easy HTML

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

Vim plugin "Easy HTML" is developed for the Web programmers. By this tool, the Web programmer just need code, for example "`div#div1.col3`" and then plugin would transform it to "`<div id = "div1" class = "col3"></div>`". Now your task is to write a program to perform similar transformation.

Here are more details about you task:

1. Handle multilevel tag.

   "`div>p>span`" means there are 3 tags and tag "`<p>`" is in the tag "`div`", tag "`<span>`' is in the tag "`p`". So, it is expanded to "`<div><p><span></span></p></div>`".

2. Every tag may have zero or one id and any amount of classes.

   A string (only consisting of letters and digits) after '`#`' is an id name.

   A string (only consisting of letters and digits) after '`.`' is a class name.

   If a tag has id and classes at the same time, you must output the id first.

   If a tag has more than one class, you must output them by the order according to the input.

   For example, "`div.aa#bb.cc.ee>p#g>span.d`" must be transformed into

   ```
   <div id="bb" class="aa cc ee">
     <p id="g">
       <span class="d"></span>
     </p>
   </div>
   ```

3. Handle parentheses.

   Use parentheses to deal with sibling relation among tags.

   For example,

   ```
   <div id="bb" class="aa cc ee">
     <p id="g1"><span class="d1"></span></p>
     <p id="g2"><span class="d2"></span></p>
     <p id="g3"><span class="d3"></span></p>
   </div>
   ```

   can be obtained by "`div.aa#bb.cc.ee>(p#g1>span.d1)(p#g2>span.d2)(p#g3>span.d3)`"

   If the input string contains parentheses, the rightmost '`)`' will be the last character of this string.

4. Handle symbol '`*`'

   At the end of a tag, you may see a suffix "`*`n", where n is positive integer. It indicates that this tag would be repeated n times.

   For example,

   `ul#id1>li.classA*3>p*2` must be transformed into

   ```
   <ul id="id1">
     <li class="classA">
       <p></p>
       <p></p>
   ```

```
        </li>
        <li class="classA">
          <p></p>
          <p></p>
        </li>
        <li class="classA">
          <p></p>
          <p></p>
        </li>
      </ul>
```

Multiple lines and indents in the samples above are used for clarify only; resulting HTML code must be printed in one line.

Input

The input consists of a string you need to process. No string has more than 120 chars and the result would not have more than 1000 chars. Tag name, class name and id only contain English letters and digits. It is guaranteed that the input string is valid.

Output

Print a string that is the result of the transformation. More details about the output format can be seen from the sample output.

Examples

standard input
`div>p>span`
standard output
`<div><p><span></span></p></div>`

standard input
`div.aa#bb.cc.ee>p#g>span.d`
standard output
`<div id="bb" class="aa cc ee"><p id="g"><span class="d"></span></p></div>`

standard input
`div.aa#bb.cc.ee>(p#g1>span.d1)(p#g2>span.d2)(p#g3>span.d3)`
standard output
`<div id="bb" class="aa cc ee"><p id="g1"><span class="d1"></span></p><p id="g2"><span class="d2"></span></p><p id="g3"><span class="d3"></span></p></div>`

standard input
`ul#id1>li.classA*3>p*2`
standard output
`<ul id="id1"><li class="classA"><p></p><p></p></li><li class="classA"><p></p><p></p></li><li class="classA"><p></p><p></p></li></ul>`

Note

All answers must be printed in one line; multiple lines in samples 3 and 4 appeared due to formatting limitations.

Problem F. Fine Study

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

In College of Fine Study i-th course has Levels from level 0 to level a_i. And at the beginning, Vasya is at Level 0 of every course, and he wants to reach the highest Level of every course.

Fortunately, there are M tutorial classes. The i-th tutoial class requires that students must reach at least Level $L1_i$ of course c_i before class begins. And after finishing the i-th tutorial class, the students will reach Level $L2_i$ of course d_i. The i-th tutoial class costs him m_i.

For example, there is a tutorial class only students who reach at least Level 5 of Math can apply. And after finishing this class, the student's Informatics will reach Level 10 if his Informatics' Level is lower than 10.

Now you task is to help Vasya to compute the minimum cost!

Input

The first line of the input file consists of two positive integers, N ($N \leq 50$) and M ($M \leq 2000$). The following line contains N integers, representing a_1 to a_N. The sum of a_1 to a_N will be not greater than 500. Each of the next M lines contain five integers, indicating c_i, $L1_i$, d_i, $L2_i$ and m_i ($1 \leq c_i, d_i \leq N$, $0 \leq L1_i \leq a_{c_i}$, $0 \leq L2_i \leq a_{d_i}$, $m_i \leq 1000$) for the i-th tutorial class. The courses are numbered from 1 to N.

Output

Output the minimum cost for achieving Vasya's target in a line. If his target can't be achieved, just output -1.

Input

Examples

standard input	standard output

3 4
3 3 1
1 0 2 3 10
2 1 1 2 10
1 2 3 1 10
3 1 1 3 10
	40

Problem G. Game of Three Thimbles

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

A gambler is playing a well known game «3 thimbles»: each round he is trying to guess what thimble has a ball inside. Exactly one thimble of 3 has a ball.

Let's consider that the probability to guess the correct thimble is p, $(0.35 \le p \le 0.55)$. For each winning round gambler is awarded by 2 lars, for each lost round he lost 1 lar.

Calculate probability that gambler will lose all his money, assumnig that he had N lars before start of the game.

Input

In the input file, two space-separated numbers are given: real p and integer N $(0.35 \le p \le 0.55$, $0 \le N \le 30)$. You can assume that p contains no more than 7 digits after decimal point.

Output

Print probability that gambler will lose all his money. Answers with absolute error 10^{-8} or better are accepted.

Examples

standard input	standard output
`0.5 0`	`1`
`0.4 3`	`0.5571891`

Problem H. Hase and Wolf

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 second
Memory limit:	128 mebibytes

The Hase runs along the circular race track on the playground every evening. One evening, The Wolf stood in the center of the circular race track. The Wolf decided to catch up with The Hase. He rushed to him; however, he could not run too far because his bad shape caused by regular smoking.

Given the maximum distance The Wolf can run, you are asked to check whether he can catch up with The Hase. Assume that the values of Wolf's and Hase's velocity are both constants, and The Wolf, The Hase and the center of the circular race track always form a straight line during the process. Note that The Wolf and The Hase can be considered as two points.

Input

The input begins with a line containing an integer T ($1 \le T \le 10^5$), which indicates the number of test cases. The following T lines each contain four integers V_r, V_w, R, and D ($0 < V_r, V_w, R, D \le 10^9$, $V_r \le V_w$). V_r is the velocity of The Hase. V_w is the velocity of The Wolf. R is the radius of the race track. D is the maximal distance The Wolf can run.

Output

For each case, output "`Run, Rabbit, Run!`" in a line if The Wolf can catch up with The Hase; otherwise output "`Nu, pogodi!`" in a line.

Example

standard input	standard output

2
1 1 1 1
11904 41076 3561 3613
	Nu, pogodi!
	Run, Rabbit, Run!

Problem I. Identification of Cockroaches

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 seconds
Memory limit:	128 mebibytes

There live n cockroaches in the campus of Kovrov STA. All cockroaches have personal numbers — positive integers from 1 to n.

Also there are several locations in the campus. Cockroaches can migrate between locations and may identify each other in there (fact that cockroach A identifies cockroach B in location $L1$ does not leads to conclusion that cockroach B identifies cockroach A in location $L1$, neither that cockroach A identifies cockroach B in some other location $L2$). Define *rating of popularity* $k_{i,l}$ of cockroach i in location l as number of different cockroaches, who can identify i in l (cockroach i itself is not included in this list).

To make communications between humanity and cockroaches easy, you must determine for each location cockroach with maximal rating of popularity.

Input

First line contains two integers m and n ($1 \le m \le 10^5$, $1 \le n \le 10^5$), where m is the quantity of identification records, and n is the total number of cockroaches. Then m lines follow, describing cockroaches identification.

Each line contains two integers a and b and a non-empty string l, consisting of no more than 12 lowercase English letters, meaning that cockroach a identifies cockroach b in the location named l ($1 \le a, b \le n$, $a \ne b$). It is guaranteed that there is no more than 15 locations and that in any two distinct lines atleast one of parameters a, b and l is different.

Output

For each location print in the new line location name and number of most popular cockroach. In case of a tie print cockroach with smallest number. Order of locations must correspond to order, in which locations appear in the input file for the first time.

Examples

standard input	standard output

`8 5`
`1 2 kitchen`
`3 2 botalka`
`3 4 room512`
`4 3 room512`
`2 3 room512`
`2 3 kitchen`
`4 2 botalka`
`5 2 kitchen`
	`kitchen 2`
	`botalka 2`
	`room512 3`

`5 2`
`1 2 room512`
`1 2 roof`
`1 2 myworkplace`
`2 1 hlebnica`
`2 1 mycomputer`
	`room512 2`
	`roof 2`
	`myworkplace 2`
	`hlebnica 1`
	`mycomputer 1`

Problem J. Juliette's Walk

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	128 mebibytes

Every afternoon, when the sun set, the beautiful girl Juliette comes out from her home, walks around the city, and returns to her home. The girl's walking path can be considered as a closed polyline.

A closed polyline is a curve specified by a sequence of points $(P_1, P_2, \ldots, P_n)$ so that the curve consists of the line segments $P_1P_2,\ P_2P_3,\ \ldots,\ P_{n-1}P_n,\ P_nP_1$. Note that two line segments may intersect, coincide or partially coincide with each other. The points $P_1,\ P_2, \ldots, P_n$ are called vertices of the polyline. While the girl is walking along the path, boy Romeo is standing at some point which is not lying on the path. During the Juliette's walking, Romeo rotates himself such that he can always see the girl directly in front of him. Romeo may sometimes rotate counter-clockwise, and sometimes rotate clockwise. If the boy stands at point P, let A_P denote the total degrees the boy rotate counter-clockwise, and B_P denote the total degrees the boy rotate clockwise. It is not difficult to see that $A_P - B_P$ is a multiple of 360, because when Juliette finishes walking, the boy faces in the same direction as when the girl starts walking. We define the rotation number for Romeo of P to be $(A_P - B_P)/360$.

Given the coordinates of n vertices $P_1, P_2, \ldots, P_n$, which specifies Juliette's walking path, you are asked to find the largest possible rotation number for Romeo among all the points that are on the X axis, but not on the girl's walking path.

Input

The input begins with a line containing an integer N ($2 \le N \le 1000$), indicating the number of vertices of the Juliette's walking path. The following N lines each contain two integers x_i and y_i ($-1000 \le x_i, y_i \le 1000$), indicating the coordinates of the i-th vertex P_i.

Output

Output the largest possible rotation number in a line.

Examples

standard input	standard output

2
1 2
2 1
	0

6
1 -1
2 -1
2 1
3 1
3 0
1 0
	0

5
1 1
-1 0
1 -1
0 1
0 -1
	2

Problem K. Kabbalah Square

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	128 mebibytes

To calculate a *digital root* of a number, you must sum the digits, that the number consists of. If the result is greater than 9, sum the digits again and repeat that action until only one digit remains in the sum.

For example, let's calculate the digital root of 123454:

```
123454: 1 + 2 + 3 + 4 + 5 + 4 = 19
19: 1 + 9 = 10
10: 1 + 0 = 1
```

So, digital root of 123454 is 1: DR(123454) $= 1$.

Given an array a_i of 9 strictly increasing positive integers, each of them does not exdeed 1000. *Kabbalah Square* of that numbers is a table V sized 9×9 cells, where cell $V_{i,j}$ contains digital root of multiplication of $a_i \cdot a_j$ ($V_{i,j} = DR(a_i \cdot a_j)$).

Your task is to write a program, that using a given table V tries to reproduce numbers $a_1, a_2, \ldots, a_9$ so that $V_{i,j} = DR(a_i \cdot a_j)$. If it is not possible, your program must inform about it. If there are several solutions, any of them is considered correct.

Input

Input contains 9 lines, each of them contains 9 numbers $V_{i,j}$ ($1 \le V_{i,j} \le 9$), separated by spaces. i-th of those lines must correspond to digital roots of $a_i \cdot a_1$, $a_i \cdot a_2$, $\ldots$, $a_i \cdot a_9$ (in case when solution exists, of course).

Output

Print 9 positive integers a_1, a_2, $\ldots$, a_9 ($a_i < a_{i+1}$ for $1 \le i \le 8$, $1 \le a_i \le 1000$) — any of possible answers to the problem ,or "`No solution`", if it is not possible to find such numbers.

Examples

standard input	standard output

`1 2 3 4 5 6 7 8 9`
`2 4 6 8 1 3 5 7 9`
`3 6 9 3 6 9 3 6 9`
`4 8 3 7 2 6 1 5 9`
`5 1 6 2 7 3 8 4 9`
`6 3 9 6 3 9 6 3 9`
`7 5 3 1 8 6 4 2 9`
`8 7 6 5 4 3 2 1 9`
`9 9 9 9 9 9 9 9 9`
	`1 2 3 4 5 6 7 8 9`

`1 2 3 4 5 6 7 8 1`
`2 4 6 8 1 3 5 7 9`
`3 6 9 3 6 9 3 6 9`
`4 8 3 7 2 6 1 5 9`
`5 1 6 2 7 3 8 4 9`
`6 3 9 6 3 9 6 3 9`
`7 5 3 1 8 6 4 2 9`
`8 7 6 5 4 3 2 1 9`
`9 9 9 9 9 9 9 9 9`
	`No solution`

Problem L. Lesson

Input file:	*standard input*
Output file:	*standard output*
Time limit:	1 seconds
Memory limit:	128 mebibytes

On the lesson, little Kolya played with plasticene. He broke the plasticene into N pieces, and put them in a line. Each piece has a volume V_i. His teacher can't stand with the disorder of the volume of the N pieces of plasticene. So he asks Kolya to merge some successive pieces so that the volume in line is symmetrical. For example, (12, 23, 23, 12), (3,1,3) and (6) are symmetrical but (13,1,31), (2, 4, 4) and (3, 2, 3, 2) are not.

As addition teacher said that merging i successive pieces into one will cost a_i penalty, and Kolya must minimize this penalty. Help him to do it. Remember, that if one piece is merged once, its impossible to use it to merge again.

Input

The first line of the input is an integer N ($0 < N \leq 5000$), indicating the number of pieces in a line. The second line contains N integers V_i — volume of each piece ($0 < V_i \leq 10^9$). The third line contains penalties — N integers p_i ($0 < p_i \leq 10^4$), it is guaranteed that $p_1 = 0$.

Output

Print one line containing the minimal penalty Kolya can get after solving his task.

Examples

standard input	standard output

5
6 2 8 7 1
0 5 2 10 20
	10