ACM-ICPC Live Archive

6884 GREAT + SWERC = PORTO

We want to have a great SWERC at Porto this year and we approached this challenge in several ways. We even framed it as a word addition problem, similar to the classic SEND+MORE=MONEY, where each letter stands for a single digit (0, 1, 2, ..., 8, 9) that makes the arithmetic operation correct. In word additions different letters cannot be assigned the same digit and the leftmost letter in a word cannot be zero (0). In particular, a single letter term cannot be zero.

$$\begin{array}{rccccc} & G & R & E & A & T \\ + & S & W & E & R & C \\ \hline = & P & O & R & T & O \end{array}$$

To solve this word addition problem we had to find positive digits for G, S and P, and digits for R, E, A, T, W, C, O, so that each letter has a different digit and the sum is correct. It turns out that, unlike the classical SEND+MORE=MONEY which has a single solution, GREAT+SWERC=PORTO has six solutions.

T=7, E=3, W=9, G=1, A=0, P=4, S=2, C=8, R=6, O=5
T=7, E=3, W=9, G=2, A=0, P=4, S=1, C=8, R=6, O=5
T=8, E=5, W=1, G=3, A=7, P=9, S=6, C=4, R=0, O=2
T=8, E=5, W=1, G=6, A=7, P=9, S=3, C=4, R=0, O=2
T=9, E=5, W=2, G=1, A=8, P=7, S=6, C=4, R=0, O=3
T=9, E=5, W=2, G=6, A=8, P=7, S=1, C=4, R=0, O=3

Having more than one solution does not make GREAT+SWERC=PORTO a good problem to solve by hand, but it is still a piece of cake for a programer. Moreover, it gives us another reason to organize SWERC again next year and, who knows, in years to come!

Given a word addition problem, compute the number of solutions (possibly zero).

Input

The input file contains several test cases, each of them as described below.

A line with an integer n, followed by n lines containing a word each with maximum length of 10 letters. The first $n - 1$ words are the terms to be added and the last line is the result. Words contain only capital letters. If words have different lengths, they must be interpreted as aligning to the right. For instance, in the SEND+MORE=MONEY problem, the D of the first word and E of the second word align with the Y of the final word. You can also assume that the size of the last word is greater than or equal to the maximum size of the preceding words, and moreover, at most ten distinct letters are involved in a word problem.

Constraints:
$3 \leq n \leq 10$
Each word has at most 10 symbols (capital letters).
A word problem has at most 10 distinct letters.

Output

For each test case, output a single line with an integer: the number of solutions of the word addition problem given as input.

Sample Input

```
3
GREAT
SWERC
PORTO
3
SEND
MORE
MONEY
5
TOO
GOOD
TO
BE
TRUE
```

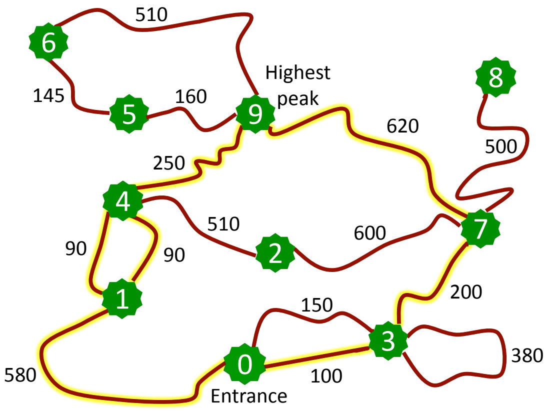
Sample Output

```
6
1
93
```

6885 Flowery Trails

In order to attract more visitors, the manager of a national park had the idea of planting flowers along both sides of the *popular trails*, which are the trails used by *common people*. *Common people* only go from the park entrance to its highest peak, where views are breathtaking, by a shortest path. So, he wants to know how many metres of flowers are needed to materialize his idea.

For instance, in the park whose map is depicted in the figure, common people make only one of the three following paths (which are the shortest paths from the entrance to the highest peak).



- At the entrance, some start in the rightmost trail to reach the point of interest 3 (after 100 metres), follow directly to point 7 (200 metres) and then pick the direct trail to the highest peak (620 metres).
- The others go to the left at the entrance and reach point 1 (after 580 metres). Then, they take one of the two trails that lead to point 4 (both have 90 metres). At point 4, they follow the direct trail to the highest peak (250 metres).

Notice that popular trails (i.e., the trails followed by common people) are highlighted in yellow. Since the sum of their lengths is 1930 metres, the extent of flowers needed to cover both sides of the popular trails is 3860 metres ($3860 = 2 \times 1930$).

Given a description of the park, with its points of interest and (two-way) trails, the goal is to find out the extent of flowers needed to cover both sides of the popular trails. It is guaranteed that, for the given inputs, there is some path from the park entrance to the highest peak.

Input

The input file contains several test cases, each of them as described below.

The first line of the input has two integers: P and T. P is the number of points of interest and T is the number of trails. Points are identified by integers, ranging from 0 to $P - 1$. The entrance point is 0 and the highest peak is point $P - 1$.

Each of the following T lines characterises a different trail. It contains three integers, p_1, p_2, and l, which indicate that the (two-way) trail links directly points p_1 and p_2 (not necessarily distinct) and has length l (in metres).

Integers in the same line are separated by a single space.

Constraints:

$2 \leq P \leq 10\,000$	Number of points.
$1 \leq T \leq 250\,000$	Number of trails.
$1 \leq l \leq 1\,000$	Length of a trail.

Output

For each test case, the output has a single line with the extent of flowers (in metres) needed to cover both sides of the popular trails.

Sample Input

```
10 15
0 1 580
1 4 90
1 4 90
4 9 250
4 2 510
2 7 600
7 3 200
3 3 380
3 0 150
0 3 100
7 8 500
7 9 620
9 6 510
6 5 145
5 9 160
4 7
0 1 1
0 2 2
0 3 10
0 3 3
1 3 2
2 3 1
1 1 1
```

Sample Output

```
3860
18
```

6886 Golf Bot

Do you like golf? I hate it. I hate golf so much that I decided to build the ultimate golf robot, a robot that will never miss a shot. I simply place it over the ball, choose the right direction and distance and, flawlessly, it will strike the ball across the air and into the hole. Golf will never be played again.

Unfortunately, it doesn't work as planned. So, here I am, standing in the green and preparing my first strike when I realize that the distance-selector knob built-in doesn't have all the distance options! Not everything is lost, as I have 2 shots.

Given my current robot, how many holes will I be able to complete in 2 strokes or less? The ball must be always on the right line between the tee and the hole. It isn't allowed to overstep it and come back.

Input

The input file contains several test cases, each of them as described below.

The first line has one integer: N, the number of different distances the Golf Bot can shoot. Each of the following N lines has one integer, k_i, the distance marked in position i of the knob.

Next line has one integer: M, the number of holes in this course. Each of the following M lines has one integer, d_j, the distance from Golf Bot to hole j.

Constraints:

$1 \leq N, M \leq 200\,000$
$1 \leq k_i, d_j \leq 200\,000$

Output

For each test case, you should output a single integer, the number of holes Golf Bot will be able to complete. Golf Bot cannot shoot over a hole on purpose and then shoot backwards.

Sample Output Explanation

Golf Bot can shoot 3 different distances (1, 3 and 5) and there are 6 holes in this course at distances 2, 4, 5, 7, 8 and 9. Golf Bot will be able to put the ball in 4 of these:

- The 1^{st} hole, at distance 2, can be reached by striking two times a distance of 1.
- The 2^{nd} hole, at distance 4, can be reached by striking with strength 3 and then strength 1 (or vice-versa).
- The 3^{rd} hole can be reached with just one stroke of strength 5.
- The 5^{th} hole can be reached with two strikes of strengths 3 and 5.

Holes 4 and 6 can never be reached.

Sample Input

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

Sample Output

```
4
```

6887 Book Club

The famous staircase in Lello bookstore, downtown Porto.

Porto's book club is buzzing with excitement for the annual book exchange event! Every year, members bring their favorite book and try to find another book they like that is owned by someone willing to trade with them.

I have been to this book exchange before, and I definitely do not want to miss it this year, but I feel that the trading should be improved. In the past, pairs of members interested in each other's books would simply trade: imagine that person A brought a book that person B liked and vice-versa, then A and B would exchange their books.

I then realized that many members were left with the same book they walked-in with... If instead of looking for pairs I looked for triplets, I could find more valid exchanges! Imagine that member A only likes member B's book, while B only likes C's book and C likes A's book. These 3 people could trade their books in a cycle and everyone would be happy!

But why stop at triplets? Cycles could be bigger and bigger! Could you help me find if it is possible for everyone to go out with a new book? Be careful, because members will not give their book without receiving one they like in return.

Given the members of the book club and the books they like, can we find cycles so that everyone receives a new book?

Input

The input file contains several test cases, each of them as described below.

The first line has two integers: N, the number of people, and M, the total number of "declarations of interest". Each of the following M lines has two integers, A and B, indicating that member A likes the book that member B brought ($0 \leq A, B < N$). Numbers A and B will never be the same (a member never likes the book he brought).

Output

For each test case, you should output '`YES`' if we can find a new book for every club member and '`NO`' if that is not possible.

Constraints

$2 \leq N \leq 10\,000$
$1 \leq M \leq 20\,000$ and $M \leq N^2 - N$.

Sample Input

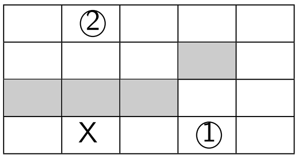
```
9 9
0 1
1 2
```

```
2 0
3 4
4 3
5 6
6 7
7 8
8 5
```

Sample Output

```
YES
```

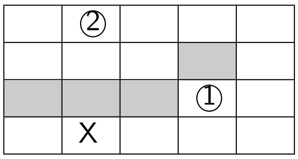
6888 Ricochet Robots



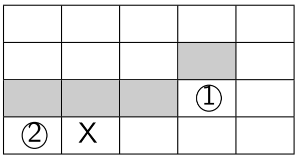
A team of up-to four robots is going to deliver parts in a factory floor. The floor is organized as a rectangular grid where each robot ocupies a single square cell. Each robot is represented by an integer from 1 to 4 and can move in the four orthogonal directions (left, right, up, down). However, once set in motion, a robot will stop only when it detects a neighbouring obstacle (i.e. walls, the edges of the factory or other stationary robots). Robots do not move simultaneously, i.e. only a single robot moves at each time step.

The goal is to compute an efficient move sequence such that robot 1 reaches a designed target spot; this may require moving other robots out of the way or to use them as obstacles for "ricocheting" moves.

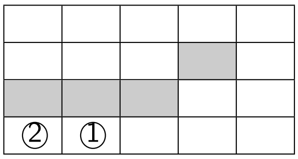
Consider the example given above, on the right, where the gray cells represent walls, X is the target location and ①, ② mark the initial positions of two robots. One optimal solution consists of the six moves described below.



① moved up.



② moved right, down and left.



① moved down and left.

Note that the move sequence must leave robot 1 at the target location and not simply pass through it (the target does not cause robots to stop — only walls, edges and other robots).

Given the description of the factory floor plan, the initial robot and target positions, compute the *minimal total number of moves* such that robot 1 reaches the target position.

Input

The input file contains several test cases, each of them as described below.

The first line contains the number of robots n, the width w and height h of the factory floor in cells, and an upper-bound limit ℓ on the number of moves for searching solutions.

The remaining h lines of text represent rows of the factory floor with exactly w characteres each representing a cell position:

'`W`' a cell occupied by a wall;

'`X`' the (single) target cell;

'`1`','`2`','`3`','`4`' initial position of a robot;

'`.`' an empty cell.

Constraints:
$1 \leq n \leq 4$
$\max(w, h) \leq 10$
$w, h \geq 1$
$1 \leq \ell \leq 10$

Output

For each test case, the output should be the minimal number of moves for robot 1 to reach the target location or '`NO SOLUTION`' if no solution with less than or equal the given upper-bound number of moves exists.

Sample Input

```
2 5 4 10
.2...
...W.
WWW..
.X.1.
1 5 4 10
.....
...W.
WWW..
.X.1.
```

Sample Output

```
6
NO SOLUTION
```

6889 City Park

Porto is blessed with a beautiful city park. The park, in the western section of the city, borders the Atlantic Ocean. It has great lawns, small forests, plenty of flowerbeds, a variety of ponds, and, in all, lots of points of interest. Porto families love the park and flock to in on weekends and holidays.

With such multitudes, it is hard work to keep the lawns in good shape. In order to control the movements of the crowd, the engineers of the municipality designed a system of paths connecting points of interest. These paths are built with large rectangular shale stones from the nearby Milhária quarry. Using sophisticated location systems, the engineers were able to lay the stones perfectly aligned with the north-south direction (and hence also with the east-west direction). Stones linking one from one point of interest to another touch each other, forming a contiguous stoned surface, and do not touch any stones belonging to any other stoned surface.

The "defend our park" movement wants to stage a demonstration in the park to publicise their cause. Since they do not want to harm the lawns, they must stage the demonstration in one of those stoned surfaces. In order to summon as many supporters as possible, but not too many, they need to find out the area of the stoned surface with largest area.

Given the locations and dimensions of stones in the park, compute the area of the stoned surface with the largest area.

Input

The input file contains several test cases, each of them as described below.

The first line contains one positive integer, N, representing the number of rectangular stones. N lines follow, each one describing the location and dimensions of a stone, by four integers, X, Y, W, H, where (X, Y) are the coordinates of the location of the lower left corner the stone, W is its length along the x-axis, and H is its length along the y-axis.

Constraints:

$0 < N \leq 50\,000$	Number of stones.
$0 < W \leq 500, \quad 0 < H \leq 500$	Dimensions of stones.

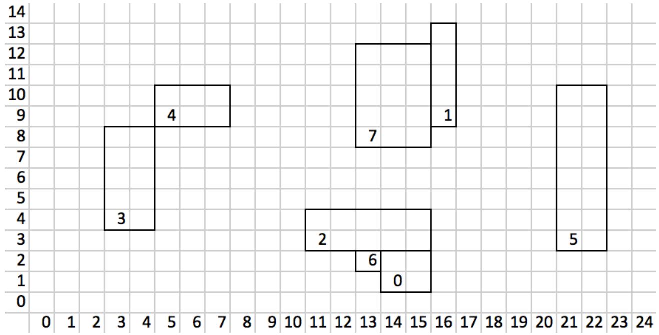
It is guaranteed that, for the given inputs, the coordinates of the stone corners can be handled using normal 32-bit signed integers, as well as the total area of any stoned surface. For every pair of distinct stones, the area of the intersection of the two rectangles that represent them in the park is zero (i.e., there are no overlaps).

Output

For each test case, write to the output a single line with an integer: the area of the stoned surface with largest area.

Sample Output Explanation

The following figure represents the configuration of stones described in the sample input.



There are 4 stoned surfaces: one made up by stones 3 and 4, on the left, with area 16; another, made up by stones 7 and 1, with area 20; a third one, below the previous, made up by stones 0, 2 and 6, with area 15; and the one on the right, made up by stone 5 only, with area 16. The largest area is 20.

Sample Input

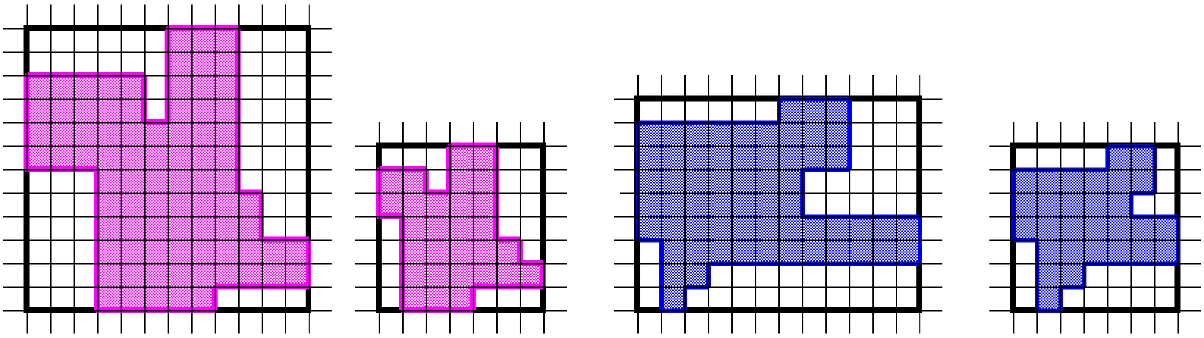
```
8
14 1 2 2
16 9 1 5
11 3 5 2
3 4 2 5
5 9 3 2
21 3 2 8
13 2 1 1
13 8 3 5
```

Sample Output

```
20
```

6890 Playing with Geometry

Dynamic geometry software can help students understand transformation geometry as it allows to visualize the effect of transformations on a shape. Alice is learning the elementary transformations – slides, flips and turns – or, more formally, translations, reflections and rotations. Today she is exploring slides and turns. She is working with simple rectilinear polygons drawn on a regular square grid. Each polygon has at most one edge per grid line and its vertices are grid points.



She knowns that: a rectilinear polygon is a polygon whose edges meet at right angles; in a simple polygon the edges meet only at their endpoints; the vertices are the points where edges meet; the region delimited by a rectilinear polygon embedded on a grid corresponds to a polyomino (formed by unit squares); a permutomino is a polyomino that has exactly one edge on all the grid lines that intersect its minimum bounding rectangle; that rectangle is actually a square (for an n-vertex permutomino, it intersects $n/2$ horizontal and vertical grid lines). Her teacher has explained how slides and turns act on point coordinates but, by that time, Alice was already thinking about an episode of "The Sympsons" in which Homer claimed that there was no more room in his brain for more information.

Now, she has to decide whether two simple rectlinear polygons, without collinear edges, can be transformed to the same permutomino by slides and turns, under some restricted rules. The two polygons are handled as independent instances, as if they were in different grids. First, she must remove the empty grid lines from the minimum bounding rectangle to obtain a permutomino. This is done by sliding some edges to the left or downwards, in such a way that the relative order of the edges is preserved (i.e., they will occur in the same order as before if we sweep the plane using a vertical or a horizontal line). Once she gets the permutomino, she can apply a rotation by 90 degrees clockwise around the center of its minimum bounding square, the number of times she wishes.

Can we give the answer to Alice's problem, for a pair of such polygons?

Input

The input file contains several test cases, each of them as described below.

The first line contains the description of the first rectilinear polygon: the number of vertices followed by their coordinates in a canonical cartesian system. The sequence of vertices is given in counterclockwise order (CCW) and starts at the leftmost vertex on the bottom horizontal edge. The last vertical edge is defined by the last vertex and the first one in the sequence. The bottom-left corner of the minimum bounding rectangle is always **(0,0)**. The next line contains a similar description for the other rectilinear polygon. The number of vertices of the two polygons may be different. (The image illustrates Sample 1.)

Constraints:

For each polygon: the number of vertices is even and between 4 and 500; the coordinates (x, y) of each vertex satisfy $0 \leq x \leq 3\,000$ and $0 \leq y \leq 3\,000$.

Output

For each test case, output a line containing the answer '`yes`' or '`no`'.

Sample Input

```
16 3 0 8 0 8 1 12 1 12 3 10 3 10 5 9 5 9 12 6 12 6 8 5 8 5 10 0 10 0 6 3 6
16 1 0 2 0 2 1 3 1 3 2 12 2 12 4 7 4 7 6 9 6 9 9 6 9 6 8 0 8 0 3 1 3
10 3 0 4 0 4 2 1 2 1 3 2 3 2 4 0 4 0 1 3 1
10 0 0 4 0 4 3 5 3 5 6 3 6 3 2 1 2 1 4 0 4
```

Sample Output

```
yes
no
```

6891 Money Transfers

Sonia is the CEO of the South Western Economic Research Consortium (SWERC). The main asset of SWERC is a group of banks spread out in several countries, which specialize in wire transfers between these countries.

Money is transferred between banks which have transfer agreements. Such agreements settle a fixed fee whenever there is a transfer between these banks. When a client wishes to transfer money to an account in a different bank, the money flows between banks with transfer agreements until it reaches the destination account. For each intermediary transaction, the client will have to pay the corresponding transfer fee.

SWERC's goal is to provide the cheapest fees for its clients, using only its banks as intermediaries, and profit from its commissions. Things were going quite well until the recent economic crisis. Due to the current situation, governments agreed to impose an extra tax on each wire transfer. Their objective is to both increase the tax income and avoid losing money to tax havens around the world. Hence, the intention is make this extra tax as large as possible (while avoiding too much unrest).

Sonia, being a savvy executive, wants to take advantage of this situation and make sure SWERC provides the cheapest way to transfer money between banks **X** and **Y** (their most frequent transfer requests). She will try to lobby the politicians so that the new fee makes this happen. She gathered data about the transfer agreements between banks (including competitors) but has no idea what should be the value of the new fee.

Can you help Sonia compute the largest fee so that SWERC can provide cheapest way to transfer money between **X** and **Y**?

Input

The input file contains several test cases, each of them as described below.

The first line consists of four space-separated integers: N P X Y, the number of existing banks, the number of transfer partnerships, and the identifiers of banks **X** and **Y**, respectively. The next P lines have three space-separated integers: a_i b_i c_i, meaning there is a partnership between banks a_i and b_i with fee c_i.

A line with an integer M, the number of banks owned by SWERC, follows. The next line contains M space-separated integers, the identifiers of these banks. X and Y are always in this list.

Constraints:

$2 \leq M \leq N \leq 1\,000$ and $1 \leq P \leq 10\,000$

$1 \leq X, Y, a_i, b_i \leq N$ and $X \neq Y$ and $a_i \neq b_i$

$1 \leq c_i \leq 1\,000\,000\,000$

Output

For each test case, the output should be on a line by itself a single integer greater than zero: the largest fee so that SWERC can provide cheapest way to transfer money between X and Y. However, if there is no value such that this happens, output '`Impossible`' instead. If the fee on each transfer can be infinitely large, output '`Infinity`'.

Sample Output 1 Explanation

If the extra fee is 4 or more, then SWERC can not provide the cheapest transaction fee. Example: if the fee is 4, SWERC provides a cost of 20, using banks 1, 3, 4, 5 and 6, in this order. However, using bank 2 as an intermediary, we can pay only 19.

Sample Input

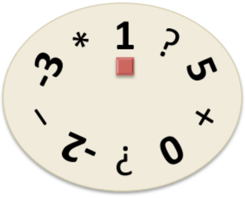
```
6 8 1 6
1 2 5
1 3 1
2 6 6
2 3 6
4 2 3
3 4 1
4 5 1
5 6 1
5
1 3 6 5 4
3 4 1 2
1 2 6
1 3 2
1 2 7
2 3 3
2
1 2
4 4 1 4
1 2 1
1 3 1
2 4 1
3 4 1
3
1 2 4
```

Sample Output

```
3
Infinity
Impossible
```

6892 The Safe Secret

One of the brightest and richest dukes of the nineteenth century built a break-in-proof room for storing his valuables and chose the lock secret code in an ingenious manner. He was so afraid of being robbed that he did not tell anyone the safe secret; he only wrote the way to obtain it on a piece of paper, to be given to his heir on his death.



1. Look at the bottom of my dukedom ring, which is now yours.
2. Write down the numbers and symbols, following a clockwise order, starting at the number closest to the ruby and leaving out the last symbol. That is the first sequence of numbers and symbols.
   Do the same starting at the next number, with respect to the clockwise order. That is the second sequence of numbers and symbols.
   Repeat this process, always starting at the next number, until you have started at all numbers. Now you have several sequences of numbers and symbols.
3. For each of those sequences of numbers and symbols, do the following.
   (a) Replace every ? by a $+$, a $-$ or a $*$ symbol.
   Do that in all possible ways to have several arithmetic expressions.
   (b) Evaluate each of those arithmetic expressions, performing the sums, the differences and the products in any order.
   Do that in all possible ways to have several values.
   (c) Select the minimum and the maximum of those values.
   (d) Write the digits of the minimum value and append to them the digits of the maximum value. That is the code of the sequence of numbers and symbols.
4. Concatenate the codes of all sequences of numbers and symbols, respecting the order in which you have obtained the sequences. That sequence of digits is the safe secret.

When the duke passed away, his son read the note and tried to find out the safe secret. The first two steps were very easy, because there were only five sequences of numbers and symbols, obtained in the following order:

1	?	5	+	0	?	-2	−	-3
5	+	0	?	-2	−	-3	∗	1
0	?	-2	−	-3	∗	1	?	5
-2	−	-3	∗	1	?	5	+	0
-3	∗	1	?	5	+	0	?	-2

Then, he moved to the third step and chose to begin with the first sequence of numbers and symbols. Difficulties started in point (a) when he realised that he could create several arithmetic expressions, such as:

$$1 + 5 + 0 + \text{-}2 - \text{-}3, \quad 1 - 5 + 0 * \text{-}2 - \text{-}3, \quad \text{and} \quad 1 * 5 + 0 - \text{-}2 - \text{-}3.$$

So, he decided to understand the remaining rules before completing this task. In point (b), he had to evaluate the arithmetic expressions. It seemed easy. The value of 1 + 5 + 0 + -2 − -3 was **7**. But how many different values could he get from 1 − 5 + 0 ∗ -2 − -3?

- If the operations were performed from left to right, ((((1 − 5) + 0) ∗ -2) − -3), the result would be **11**.
- If the operations were performed from right to left, (1 − (5 + (0 ∗ (-2 − -3)))), the result would be **-4**.
- If the first difference and the product were performed first, (1 − 5) + (0 ∗ -2) − -3, the result would be **-1**.
- And there were so many other alternatives!

Almost in despair, he concluded that he had to obtain a huge number of values in the third step. Fortunately, the last rules were actually simple. If **-4** was the minimum of the values obtained from the first sequence and **11** was the maximum, the code of the first sequence would be **411**. Besides, if the second sequence code was **512**, the third sequence code was **613**, the fourth sequence code was **714**, and the fifth sequence code was **815**, the safe secret would be **411512613714815**.

Although the duke's son spared no effort in finding the secret, he has never achieved that goal. In fact, no one has managed to open the safe so far. Now that the palace will be transformed into a museum, could you help unveiling the treasure?

Given the sequence of numbers and symbols obtained from the dukedom ring, starting at the number closest to the ruby, following a clockwise order, and including the last symbol, the goal is to find out the safe secret. It is guaranteed that, for the given inputs, any value obtained by the process described above fits in a normal signed 64 bit integer.

Input

The input file contains several test cases, each of them as described below.

The first line of the input has one positive integer, k, which is the number of pairs (*number*, *symbol*) that form the sequence.

The following line contains $2k$ elements, $n_1, s_1, n_2, s_2, \ldots, n_k, s_k$, separated by a single space, where n_i denotes a number and s_i denotes a symbol that is $+$, $-$, $*$, or ? (for every $i = 1, 2, \ldots, k$).

Constraints:

$2 \le k \le 200$	Number of pairs (*number*, *symbol*) that form the sequence.
$-9 \le n_i \le 9$	Number in the sequence.

Output

For each test case, the output has a single line with the safe secret.

Sample Input

```
5
1 ? 5 + 0 ? -2 - -3 *
```

Sample Output

```
914710203014163336
```

6893 The Big Painting

Galerie de Vues de la Rome Moderne, Panini (1759). What S.W.E.R.C. had in mind when he tried to merge his paintings.

Samuel W. E. R. Craft is an artist with a growing reputation. Unfortunately, the paintings he sells do not provide him enough money for his daily expenses plus the new supplies he needs. He had a brilliant idea yesterday when he ran out of blank canvas: ”Why don’t I create a gigantic new painting, made of all the unsellable paintings I have, stitched together?”. After a full day of work, his masterpiece was complete.

That’s when he received an unexpected phone call: a client saw a photograph of one of his paintings and is willing to buy it now! He had forgotten to tell the art gallery to remove his old works from the catalog! He would usually welcome a call like this, but how is he going to find his old work in the huge figure in front of him?

Given a black-and-white representation of his original painting and a black-and-white representation of his masterpiece, can you help S.W.E.R.C. identify in how many locations his painting might be?

Input

The input file contains several test cases, each of them as described below.

The first line consists of 4 space-separated integers: h_p w_p h_m w_m, the height and width of the painting he needs to find, and the height and width of his masterpiece, respectively.

The next h_p lines have w_p lower-case characters representing his painting. After that, the next h_m lines have w_m lower-case characters representing his masterpiece. Each character will be either ‘x’ or ‘o’.

Constraints:

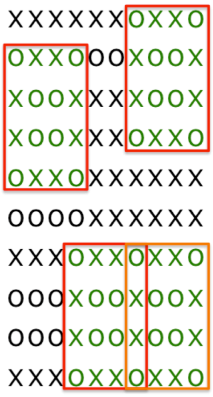
$1 \leq h_p, w_p \leq 2\,000$
$1 \leq h_m, w_m \leq 2\,000$
$h_p \leq h_m$
$w_p \leq w_m$

Output

For each test case, output a single integer representing the number of possible locations where his painting might be, on a line by itself.

Sample Output Explanation

The painting could be in four locations as shown in the following picture. Two of the locations overlap.



```
xxxxxxoxxo
oxxoooxoox
xooxxxxoox
xooxxxoxxo
oxxoxxxxxx
ooooxxxxxx
xxxoxxoxxo
oooxooxoox
oooxooxoox
xxxoxxoxxo
```

Sample Input

```
4 4 10 10
oxxo
xoox
xoox
oxxo
xxxxxxoxxo
oxxoooxoox
xooxxxxoox
xooxxxoxxo
oxxoxxxxxx
ooooxxxxxx
xxxoxxoxxo
oooxooxoox
oooxooxoox
xxxoxxoxxo
```

Sample Output

```
4
```