Ace. ModuloFourDivisor

A quadruplet of non-negative integers (a, b, c, d) is called a **divisor quadruplet** if there exists at least one positive integer N that satisfies the following four conditions:

- N has exactly a divisors x such that $x \equiv 0 \pmod 4$.
- N has exactly b divisors x such that $x \equiv 0 \pmod 4$.
- N has exactly c divisors x such that $x \equiv 0 \pmod 4$.
- N has exactly d divisors x such that $x \equiv 0 \pmod 4$.

You are given four arrays $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$. Find out the number of divisor quadruplets (a, b, c, d) such that $a \in \mathcal{A}, b \in \mathcal{B}, c \in \mathcal{C}, d \in \mathcal{D}$.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

There are four parts in each test case, all parts are in same format: first line an integer n_i, second line n_i integers in the range $[0, 10^{18}]$. These arrays appear in the order $\mathcal{A}, \mathcal{B}, \mathcal{C}$ and $\mathcal{D}$. $1 \le n_i \le 50$ for all $1 \le i \le 4$.

Output

For each test case, output the desired number in a line.

Sample Input

```
1
1
1
1
1
1
1
0

1
0
1
0
1
0
1
0

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

Sample Output

```
1
0
1
```

Blue. SpaceshipEvacuation

Manao is an engineer of a new spaceship. He is responsible for crew safety, particularly during a possible evacuation. The spaceship consists of N units numbered from 0 to $N-1$. The units are connected by passages, but during evacuation, moving through them is too slow and therefore is prohibited. Some pairs of units are connected by tunnels which contain emergency cabins. There are several emergency cabins at each end of each tunnel. A cabin is designed for a single person and may only be used once for security reasons. A cabin at one end of a tunnel can only be used to reach the other end of that same tunnel.

The tunnel network has a special layout. Consider a sequence of units $U_0, U_1, \ldots, U_K$ with $K \geq 3$ and $U_0 = U_K$. If all $U_0, U_1, \ldots, U_{K-1}$ are pairwise distinct and for each i, $0 \leq i < K$, U_i and U_{i+1} are connected by a tunnel, this sequence is called a cycle. A cycle is called canonical if $U_0 < U_i$ for $1 \leq i < K$ and $U_1 < U_{K-1}$. Each unit in the spaceship will be a part of at most one canonical cycle. The tunnel network is given as m lines with each line containing four integers A, B, C, D. This means that there is a tunnel between units A and B and there are C emergency cabins in the tunnel from unit A's side and D emergency cabins from unit B's side.

The crew of the spaceship consists of $crewSize$ members. Each of them might be in any of the N units when the evacuation is announced. Unit 0 is connected to the rescue boat, so every person reaching this unit is considered evacuated. The distribution of emergency cabins within the tunnels is called an evacuation plan. An evacuation plan is called acceptable if there exists a way to evacuate the whole crew for any possible distribution of crew members over the units at the moment when the evacuation is announced. The current evacuation plan might not be acceptable. Manao may demand a number of additional emergency cabins at each end of each tunnel, but he is not allowed to change the location of emergency cabins that are already present in the spaceship. Return the minimum total number of emergency cabins Manao has to add to obtain an acceptable evacuation plan from the current one. If it is impossible, output -1.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line contains three integers N, m and $crewSize$. Next m lines contains four integers in each line. $2 \leq N \leq 50; 1 \leq m \leq 50; 0 \leq A, B \leq N-1; 0 \leq C, D \leq 100000; A \neq B; 1 \leq crewSize \leq 100000$.

Output

For each test case, please output the desired number in a line.

Sample Input

```
3 2 5
0 1 5 3
2 1 0 0

3 2 5
0 1 0 2
0 2 0 4

4 4 6
0 1 0 6
3 2 3 1
2 1 0 1
3 1 2 2

10 11 110
0 1 11 101
1 2 0 100
2 3 20 100
3 4 0 107
4 1 66 0
3 5 104 2
4 6 82 0
5 7 25 25
7 8 14 42
8 9 0 94
9 5 17 92

3 1 1
0 1 0 0
```

Sample Output

```
7
4
6
376
-1
```

Combo. RabbitProgramming

Rabbits often feel lonely, so they enjoy participating in programming contests together.

Rabbit Iris is the head coach of her university's programming team. The big annual contest is going to be held next month, so she decided to hold a preliminary contest to help her decide who to put in the team.

The preliminary contest is now over, and the submissions are being reviewed. You are given an array of integer `points`, and an array of strings `standings`. Each element of points represents a single problem from the contest. For the j-th problem:

- If `points[`j`]` is positive, then all submissions for this problem have been reviewed, and the point value of the problem is `points[`j`]`. The j-th character of the i-th element of `standings` is '`Y`' if rabbit i correctly solved the problem, or '`N`' if he did not.
- If `points[`j`]` is negative, then none of the submissions for this problem have been reviewed yet, and the point value of the problem is $-$`points[`j`]`. The j-th character of the i-th element of standings is '`Y`' if rabbit i submitted a solution (which may or may not be correct) for this problem, or '`N`' if he did not.

A rabbit's score is the sum of the point values for the problems which he solved correctly. Once all the submissions are reviewed, the rabbits will be ranked according to their scores. Rabbits with higher scores will be ranked higher than rabbits with lower scores. If two rabbits have the same score, then the lower-numbered rabbit will be ranked higher. The top `qualified` ranking rabbits will be qualified to be members of the team. Among them, Iris will arbitrarily select `selected` rabbits to actually be in the team. If you consider all the possible scenarios for the problems which have not yet been reviewed, how many different teams are possible?

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line contains four integers $n, m, qualified, selected$, $(1 \leq n, m \leq 50)$. Next n integers represent the array of `points`. Then next m strings represent the array of `standings` with each string has length exactly n, and each character will be either `Y` or `N`. $1 \leq selected \leq qualified \leq m$. All numbers in `points` will be between -100000 and 100000, inclusive.

Output

For each test case, please output the desired number.

Sample Input

```
2 4 3 2
1 -10
NY
YN
YN
YN
3 5 4 2
-250 -500 -1000
YYY
YNY
YYN
YYN
YNN
12 10 4 3
5 -12 5 -15 10 -20 3 -25 7 -32 21 -45
YYYYYYYYYNYY
YYYNYYYYYNNN
YYYNYNYYNNYN
YYYYYNYYYYNN
YYNNYNYNYYNN
YYYNNNYYNNNN
YYNNNNYYNNNN
NNYNYYYNYNNN
NNNNNNYYYNNN
YYYNNNYYYNNN
```

Sample Output

```
5
10
99
```

Digger. RooksParty

The black and white chess rooks were bored, so they decided to invite their colorful friends to a party. However, as the evening progressed, many pairs of rooks of different colors started arguing and threatening each other.

To prevent a massacre, you now need to place all the rooks in such a way that no two rooks of different colors attack each other.

You are given the dimensions of the chessboard: ints rows and columns. You are also given an array of integers `counts`, where `counts[`i`]` is the number of rooks of the i-th color you have.

Compute and return the value (X mod $1,000,000,009$), where X is the number of valid arrangements of all the given rooks on the given chessboard. No square of the chessboard may contain more than one rook. Rooks of the same color are undistinguishable.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line contains three integers r, c, n, $1 \leq r, c \leq 30; 1 \leq n \leq 10$, where r is the number of rows and c is the number of columns. Next n positive integers represent the `counts` array. The sum of all numbers in `counts` will not exceed $r \times c$.

Output

For each test case please output the desired number.

Sample Input

```
2 3 2
1 1
5 2 1
3
5 2 3
1 1 1
8 8 8
1 1 1 1 1 1 1 1
4 2 2
3 1
```

Sample Output

```
12
120
0
625702391
8
```

(Almost empty page.)

Eye. CuttingGlass

You have a machine that cuts glass panes using a robotic arm with a diamond point at the end.

The input to the machine is a rectangular piece of glass with width W and height H. The machine has a coordinate system with point $(0, 0)$ in one corner of the glass pane and (W, H) in the opposite corner. In the beginning, the diamond point is positioned at $(start_x, start_y)$. Then the machine executes a given program.

The program is given as a string. Each character in the program describes one movement of the diamond point: 'L' decreases its x-coordinate, 'R' increases its x-coordinate, 'U' decreases its y-coordinate and 'D' increases its y-coordinate by 1. During all movements, the diamond point cuts the glass.

Once the cutting is over, the glass may fall apart into multiple pieces. Compute and return their count.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

There are only two lines, first line contains four integers $W, H, start_x, start_y$, $(1 \leq W, H \leq 1000; 0 \leq start_x \leq W; 0 \leq start_y \leq H)$. Next line contains a program string. The length of program will be no greater than 2500. The program will be such that the diamond point never leaves the glass pane, but it may touch the boundary and even cut along the boundary.

Output

For each test case, please output the desired number.

Sample Input

```
100 100 50 50
ULDR
10 10 3 4
UDUDUDUDUDU
3 3 0 0
RDDDUUURDDDUUURDLLLRRRDLLL
5 3 5 3
LULLULLU
```

Sample Output

```
2
1
9
2
```

(Almost empty page.)

Force. MallSecurity

The first mall of the kingdom is about to be inaugurated in a few days. The king wants to make sure that the mall is highly secured.

The mall has N floors, numbered 1 to N. Each floor has stations which allow people to enter or leave the floor to which the station belongs. All escalators in the mall begin and end at stations of adjacent floors. To make movement of people easier, super-escalators connect stations in floor 1 to stations in floor N. Each escalator or super-escalator can be used to go upwards as well as downwards. If the i-th floor has K_i stations then the stations are numbered from 1 to K_i. The escalators and super-escalators are constructed in such a way that a person can reach any station from any other station using them.

The king wants to have as many guards in the mall as possible to make it secure. Guards can only be placed at stations and at most 1 guard can be placed at a station. Moreover, the people of the kingdom become panicky if they see more than one guard at a time. Hence, there should be no such escalator (or super-escalator) such that guards are placed at both its end stations.

You are given a list of escalators and super-escalators. Every escalator (or super-escalator) is of the form "ABC". If C is less than N, the String represents an escalator from station A of floor C, to station B of floor $C + 1$. If C is equal to N, then the String represents a super-escalator from station A of floor C $(= N)$, to station B of floor 1. Help the king by returning the maximum number of guards he can place in the mall.

Input

There are multiple test cases in the input file terminated by EOF. For each test case:

First line contains two integers N and M. Next M lines contains description of escalators (or super-escalator) each, in the form mentioned above.

Output

For each test case, please output the desired number.

Sample Input

```
10 10
1 1 1   1 1 2   1 1 3   1 1 4
1 1 5   1 1 6   1 1 7   1 1 8
1 1 9   1 1 10

11 10
1 1 1   1 1 2   1 1 3   1 1 4
1 1 5   1 1 6   1 1 7   1 1 8
1 1 9   1 1 10

10 46
1 1 7   1 2 9   2 1 8   1 2 6
1 1 8   1 2 3   1 2 2   2 2 4
1 1 1   2 1 2   3 2 3   1 1 5
2 1 1   4 1 7   1 1 10   3 2 5
1 2 5   3 3 1   3 2 8   3 1 2
1 1 3   4 4 2   2 4 6   4 2 5
2 3 3   6 4 1   5 2 8   1 3 6
1 3 7   4 3 8   1 3 8   5 2 3
4 2 8   2 6 7   1 3 9   1 1 4
6 1 1   2 3 1   5 1 5   6 1 8
5 2 2   3 2 10   3 3 9   1 5 2
4 1 1   1 5 10
```

Sample Output

```
5
6
25
```

Good. RabbitPuzzle

Taro and Hanako are playing Rabbit Puzzle. There are three rabbits and three nests on a line. You are given their initial positions, and an their nest positions.

They must perform the following routine exactly K times:

1. Choose two different rabbits A and B, located at points a and b, respectively.
2. A jumps over B and lands at point $2 \times b - a$.
3. The jump is not allowed if another rabbit is already at point $2 \times b - a$.
4. The jump is also not allowed if A jumps over more than one rabbit.
   `A-----C--B-------->A` is NOT allowed.

The goal of the puzzle is to have every rabbit be in a nest after the K routines. Note that the i-th rabbit doesn't necessarily have to be in the i-th nest. Return the number of ways to solve this puzzle, modulo $1,000,000,007$. Two ways are considered different if at least one jump is different.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, it consists 7 integers sequentially. They are $r_1, r_2, r_3, n_1, n_2, n_3, K$ respectively. ($-10^{18} \leq r_1 \leq r_2 \leq r_3 \leq 10^{18}; -10^{18} \leq n_1 \leq n_2 \leq n_3 \leq 10^{18}); 1 \leq K \leq 100$.)

Output

For each test case please output the desired number.

Sample Input

```
0 5 8 0 8 11 1
0 5 8 0 8 11 3
0 5 8 0 8 11 2
5 8 58 13 22 64 58
0 1 2 1 2 3 100
5 8 58 20 26 61 58
67 281 2348 235 1394 3293 83
-1000000000000000000
999999999999999998
999999999999999999
-1000000000000000000
999999999999999999
1000000000000000000
5
```

Sample Output

```
1
5
0
0
0
537851168
167142023
29
```

Huff. FunctionalEquation

f is a function from integers to integers. In other words, f is defined over integers, and $f(x)$ is an integer for all integers x. You are given an integer C. f is called C-beautiful if the following equality is satisfied for all integers x:

$$f(2f(x) - x + 1) = f(x) + C$$

Return the minimal possible value of the following formula when f is C-beautiful:

$$\sum_{i=0}^{N-1} |f(x_i) - y_i|$$

Use the following recursive definitions to generate the sequences x and y:

- $x_0 =$`xzero`.
- For all $1 \leq i \leq N-1$, $x_i = (x_{i-1} \times$ `xprod` + `xadd`) mod `xmod`.
- $y_0 =$`yzero`.
- For all $1 \leq i \leq N-1$, $y_i = (y_{i-1} \times$ `yprod` + `yadd`) mod `ymod`.

Input

There are multiple test cases in the input file terminated by EOF. For each test case, there are 10 integers appear consecutively. They are C, N,`xzero`, `xprod`, `xadd`, `xmod`, `yzero`, `yprod`, `yadd`, `ymod`, respectively. ($1 \leq C \leq 16; 1 \leq N \leq 10000; 1 \leq$ `xmod`, `ymod` $\leq 10^9$). Other variables will within the range of `xmod` and `ymod`.

Output

For each test case, please output the desired number.

Sample Input

```
3 10 0 1 1 456 1 1 1 456
16
10000
654816386
163457813
165911619
987654321
817645381
871564816
614735118
876543210
```

Sample Output

```
5
3150803357206
```