ICPC SOUTH PACIFIC REGIONALS

OCTOBER 1, 2023

Contest Problems

A: Attention Stones
B: Bingo Boards
C: Cave Escape
D: Door to Treasure
E: Eating Bacteria
F: Flawed LLM
G: Gem Machine
H: Hiring an Accountant
I: Investigator of Crashes
J: Jay's Son Ja$on's Family Tree
K: Killing Monsters
L: Lonely Wizard

This contest contains twelve problems. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters $(A, B, \ldots, Z)$.
- Lowercase letters are the lowercase English letters $(a, b, \ldots, z)$.

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

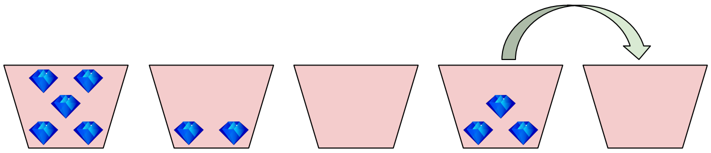
$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Problem A
Attention Stones
Time limit: 5 seconds

The newest craze to hit the South Pacific is a game called Attention Stones. In Attention Stones, there are N bins labeled 1 to N from left to right. There may be stones in each of the bins. This is a two-player game with alternating turns: Aaron's turn first, then Bertha next, then Aaron, and so on.

On each player's turn, they must find the rightmost nonempty bin (say bin B) and select between 1 and K stones, inclusive, from that bin. If that bin was the rightmost bin ($B = N$), then these stones are removed from the game. Otherwise, these stones are placed into bin $B + 1$.

In the example below, if $K = 2$, then the player must move either 1 or 2 stones from bin 4 to bin 5. However, if $K = 5$, then they may move 1, 2, or all 3 stones from bin 4 to bin 5.



The game is won by the player who removes the final stones from the game. Assuming both players play optimally, who wins?

Input

The first line of input contains two integers N ($1 \le N \le 200\,000$), which is the number of bins, and K ($1 \le K \le 10^{18}$), which is the maximum number of stones that may be moved on each move.

The next line contains N integers $b_1, b_2, \ldots, b_N$ ($0 \le b_i \le 10^{18}$), which is the number of stones in each bin at the start of the game. There is at least one i such that $b_i > 0$.

Output

Display the name of the winner assuming both players play optimally.

Sample Input 1	Sample Output 1
5 5 5 2 0 3 0	Aaron

Sample Input 2	Sample Output 2
8 12 3 141 592 653 589 793 238 462	Aaron

Sample Input 3	Sample Output 3
4 8 2023 0 0 0	Bertha

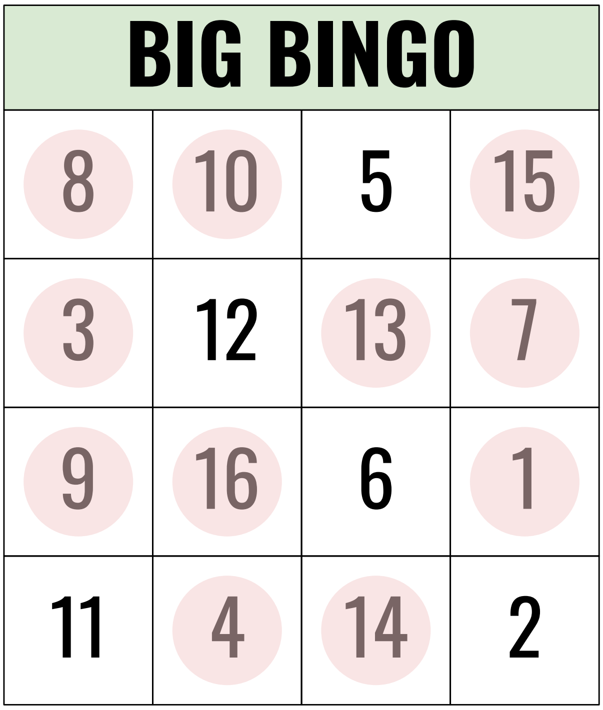
This page is intentionally left (almost) blank.

Problem B
Bingo Boards
Time limit: 1 second

John loves Bingo! At his local pub, they play a special version of Bingo every Thursday called Big Bingo.

In Big Bingo, you have an $N \times N$ grid of numbers. Each number from 1 to N^2 appears exactly once in the grid. The host of the event calls out numbers one at a time, and for each one, you mark the cell containing it. Your board wins if every cell in some row or some column is marked. Unlike normal Bingo, you cannot win with diagonals.

Things are getting very exciting this week: as John looks down, he realizes that no matter which of the five remaining numbers are called, he will win.



A partially marked board is *on the verge* if (1) it is not a winning board based on the already marked numbers and (2) marking any of the unmarked numbers will make it a winning board.

John wonders how common this is. Given N and k, find an $N \times N$ board that has exactly k marked cells and is on the verge, or state that one doesn't exist.

Input

The input consists of a single line containing two integers N ($1 \leq N \leq 200$), which is the size of the board, and k ($0 \leq k \leq N^2$), which is the desired number of marked cells.

Output

If it is impossible to construct a $N \times N$ board containing k marked cells that is also on the verge, display `IMPOSSIBLE`.

Otherwise, display `POSSIBLE`, then display a valid board showing such a marking. The board must be an $N \times N$ grid of dots (‘`.`’) and hashes (‘`#`’) without any spaces separating characters on the same line. The jth character of the ith row denotes whether the cell (i, j) is marked (`#`) or not (`.`).

If there are multiple valid solutions, any will be accepted.

Sample Input 1

```
4 11
```

Sample Output 1

```
POSSIBLE
##.#
#.##
##.#
.##.
```

Sample Input 2

```
10 65
```

Sample Output 2

```
IMPOSSIBLE
```

Sample Input 3

```
3 6
```

Sample Output 3

```
POSSIBLE
.##
##.
#.#
```

Sample Input 4

```
1 0
```

Sample Output 4

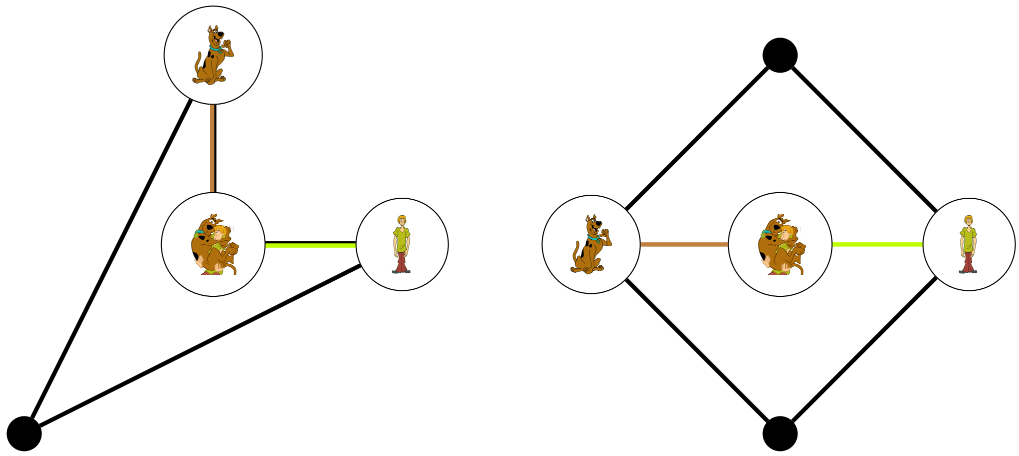
```
POSSIBLE
.
```

Problem C
Cave Escape
Time limit: 10 seconds

Scooby-Doo and Shaggy are exploring an abandoned castle, when they stand on a trapdoor and fall into in a cave. The cave has n corners and has straight stone walls joining these corners. There seems to be no way out, so they decide to check the corners carefully to look for any crack that they could escape through. They are each checking a corner when they hear a low growl.

"Zoinks, Scoob!" "Ruh-roh, Raggy!"

Is it a monster or just their stomachs rumbling? They aren't sticking around to find out! Trembling in fear, they need to run to each other, trying to huddle together as quickly as possible.



Shaggy and Scooby could be at any of the N corners in the room (including at the same corner). For each pair of corners, find the minimum total distance they would have to run in order to reach each other if they started at those two corners.

Input

The first line of the input contains two integers N ($3 \le N \le 200$), which is the number of corners of the cave.

The next N lines each contain two integers X ($-10^6 \le X \le 10^6$) and Y ($-10^6 \le Y \le 10^6$), which are the coordinates of each corner of the cave in metres, provided in cyclic order anticlockwise around the cave. The polygon defining the cave is *simple*, that is, its vertices are distinct and no two edges of the polygon intersect or touch, except that consecutive edges touch at their common corner. In addition, no two consecutive edges are collinear.

Output

For each of the N^2 pairs of corners, display the minimum total distance that Shaggy and Scooby-Doo have to run, in metres.

The distances should be displayed in order: the first N values are the distances from the first corner in the input to all others (in the order of the input), the next N values are the distances from the second corner in the input to all others (in the order of the input), and so on.

Each distance should have an absolute or relative error of less than 10^{-6}.

Sample Input 1

```
4
10 0
0 10
-10 0
0 -10
```

Sample Output 1

```
0.000000000 14.142135624 20.000000000 14.142135624
14.142135624 0.000000000 14.142135624 20.000000000
20.000000000 14.142135624 0.000000000 14.142135624
14.142135624 20.000000000 14.142135624 0.000000000
```

Sample Input 2

```
4
10 0
0 0
0 10
-10 -10
```

Sample Output 2

```
0.000000000 10.000000000 20.000000000 22.360679775
10.000000000 0.000000000 10.000000000 14.142135624
20.000000000 10.000000000 0.000000000 22.360679775
22.360679775 14.142135624 22.360679775 0.000000000
```

Problem D
Door to Treasure
Time limit: 10 seconds

Elmo is hunting for pirate treasure. The treasure is locked behind a door which requires a special key to be opened. To obtain this key, Elmo must solve a pirate puzzle involving N keys and M chests. The keys are labelled from 1 to N and the chests are labelled from 1 to M. The Nth key is the special key that will open the door to the treasure.

Each chest contains a single key. Also, each chest requires a non-empty set of keys to unlock it. Once the chest is unlocked, Elmo gets the key inside and can use it as many times as they like in the future. It is possible for two chests to contain duplicate keys. Elmo may start with 0 or more keys in their pocket that can be used to open chests. These pocket keys were previously acquired during Elmo's prior swashbuckling adventures.

Since Elmo does not want to waste time, they want to find an efficient strategy to open the door to the treasure. A strategy is a (possibly empty) set of chests that can be arranged into at least one *valid order* specifying the sequence of chests to unlock. To be a valid order, you must obtain the Nth key at some point and every chest in the sequence must be unlockable using the keys obtained prior to unlocking it. An *efficient* strategy is one in which no chest in the set of chests can be removed to give a smaller strategy. Note that there may be multiple efficient strategies possibly of different sizes. Help Elmo find an efficient strategy to open the door to the treasure.

Input

The first line of input contains two integers, N ($1 \leq N \leq 200\,000$) and M ($0 \leq M \leq 200\,000$), which are the number of keys and chests, respectively.

The next M lines describe the chests. The ith line describes the ith chest. Each starts with two integers c_i ($1 \leq c_i \leq N$), which is the key contained in the chest and u_i ($1 \leq u_i \leq N$), which is the number of keys required to unlock the chest. The rest of the line contains u_i integers each in the inclusive range from 1 to N. These describe the keys required to unlock the ith chest. It is guaranteed that there are no duplicate keys required to unlock the chest.

The sum of u_i over all chests is at most $400\,000$.

Any keys which are not contained in a chest are the keys Elmo starts with in their pocket. Note that these may appear as requirements to unlock a chest.

Output

Display a line containing a single integer k, which is the number of chests in an efficient strategy to open the door to the treasure. Then, display a line containing k integers which are the (1-indexed) indices of chests in a valid order. If there are multiple efficient strategies or valid orders, any will be accepted. If there is no strategy, display a line containing `-1` instead.

Sample Input 1

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

Sample Output 1

```
1
1
```

Sample Input 2

```
5 6
2 1 1
3 1 1
5 2 3 4
3 1 5
4 1 5
5 2 2 3
```

Sample Output 2

```
3
1 2 6
```

Sample Input 3

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

Sample Output 3

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

Sample Input 4

```
1 2
1 1 1
1 1 1
```

Sample Output 4

```
-1
```

Sample Input 5

```
10 0
```

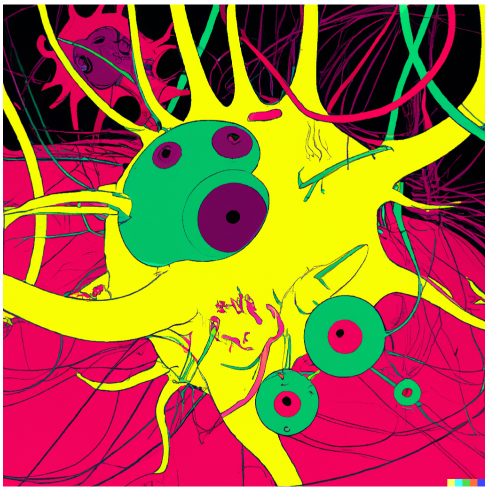
Sample Output 5

```
0
```

Problem E
Eating Bacteria
Time limit: 7 seconds

You want to become the biggest bacterium around. You live in a Petri dish with N other bacteria. Every bacterium has an integer size. In a single hour, you can hunt down and consume a single bacterium. You can only consume bacteria whose sizes are strictly smaller than your own. Consuming a bacterium will increase your size. If your size is s_1 and the consumed bacterium has size s_2 then your size will increase to $s_1 + \lceil\sqrt{s_2}\rceil$. That is, your size will increase by the rounded up square root of the size of the consumed bacterium. Once a bacterium is consumed, it is gone forever and cannot be consumed again later.

Note that only you can consume bacteria. The other bacteria cannot consume each other.



Given the N bacteria in the Petri dish, your starting size S, and a number of hours H, determine the maximum size you can reach after H hours.

Input

The first line of input contains three integers, N ($1 \le N \le 300\,000$), S ($1 \le S \le 10^6$), and H ($1 \le H \le N$), which describe the number of bacteria, your starting size, and the number of hours, respectively. The next line contains N integers each in the inclusive range from 1 to 10^6. These describe the sizes of the bacteria in the Petri dish other than yourself. There may be multiple bacteria with the same size.

Output

Display the maximum size you can achieve after H hours.

Sample Input 1	Sample Output 1
3 2 1 1 2 3	3

Sample Input 2	Sample Output 2
5 3 4 2 4 1 6 9	13

Sample Input 3	Sample Output 3
6 3 6 1 9 10 20 30 40	4

Sample Input 4

```
5 2 5
1 2 1 2 1
```

Sample Output 4

```
9
```

Problem F
Flawed LLM
Time limit: 3 seconds

You have been working for Businessy Inc. for almost a year now and the company is holding its large annual conference. You need to travel from your home city to the city where the conference is being held but you would like to stop at other cities along the way to do some sightseeing. You have identified N cities that you think are interesting enough to visit (city 1 is your home city and city N is the conference city). Each city is located at a unique place on the XY-plane.

To cut costs, Businessy Inc. has switched to using an LLM (Large Language Model) to analyze and approve expense reports. Luckily, you discovered a flaw in this system due to its biased training data. Your travel expenses are automatically approved by the LLM as long as all of your travel is along straight lines from city to city and, while traveling along the straight lines, your distance to city N is non-increasing.

What is the maximum number of extra cities (not including your home city or the conference city) that you can visit while traveling to the conference and still having your expense report approved?

Input

The first line of the input contains a single integer N ($2 \leq N \leq 1\,000$), which is the number of interesting cities.

The following N lines each contain two integers X ($-10^6 \leq X \leq 10^6$) and Y ($-10^6 \leq Y \leq 10^6$), which are the coordinates of each city.

No two cities are at the same point.

Output

Display the maximum number of extra cities you can visit while still having your expense report approved.

Sample Input 1	Sample Output 1
4 0 0 10 0 0 10 10 10	1

Sample Input 2	Sample Output 2
7 -3 0 -2 0 -1 0 1 0 2 0 3 0 0 0	2

Problem G
Gem Machine
Time limit: 9 seconds

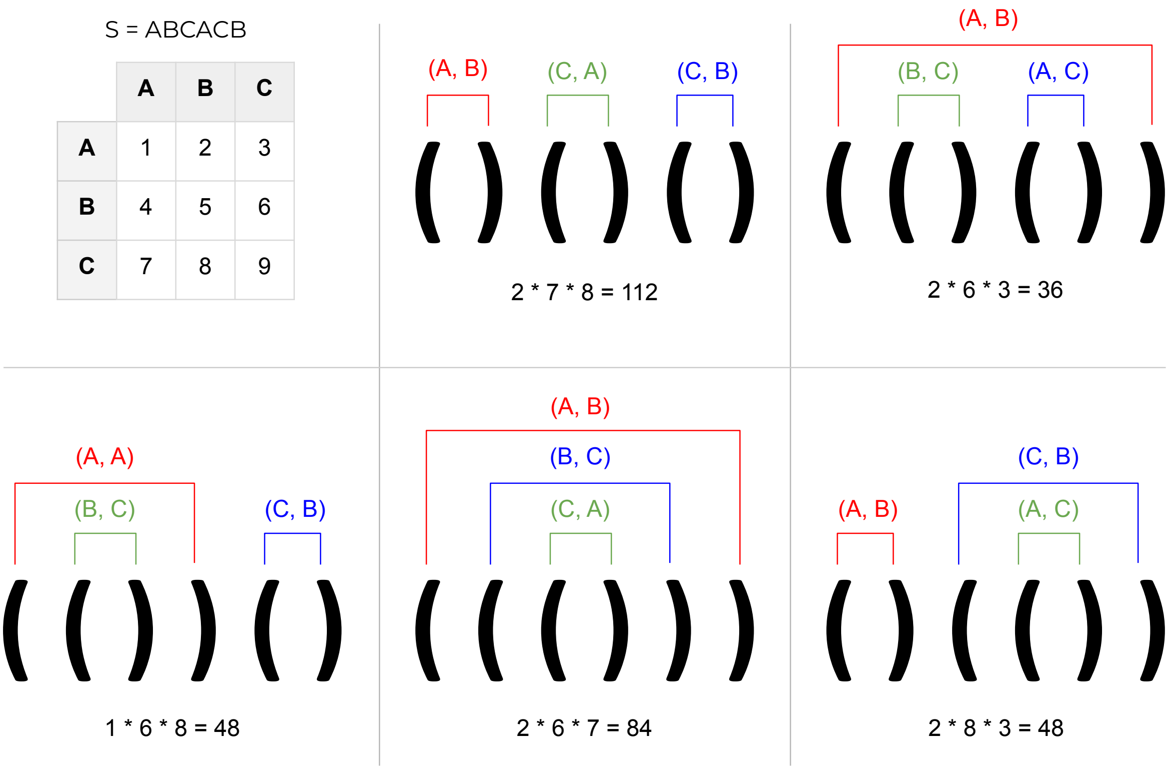
Lucy is working on an alien computer chip. The chip can be described by a string S of length N. The string contains A types of letters. Since A is at most 26, they are denoted by the first A letters of the English alphabet.

The chip takes a bracket string of length N as input and produces a valuable gem as output. A bracket string comprises only the letters "`(`" and "`)`" and must be properly nested. Every opening bracket is paired with its closing bracket.

The value of the gem is the product of bracket pair values. The value of a bracket pair is V_{S_i,S_j} where i, j are the indexes of the opening and closing brackets respectively and S_i, S_j are the letters at those indexes in S.

The overall value of the chip to Lucy is the sum over all the values of the gems it can produce. This is because it will only produce an output gem once for each valid bracket string.

For example, if S = `ABCACB` and V is the matrix in the top-left corner of the image, there are exactly 5 bracket strings of length 6. The 5 gems produced (values of 112, 36, 48, 84, and 48) are shown in the image.



Lucy has figured out a way to modify exactly one letter in the string S. Your job is to write a program to determine for each combination of index and letter, what the overall value of the chip would be if Lucy modified the chip so that it had that letter at that index. It is possible that some of these modifications do not change S.

Input

The first line of input contains two integers, N ($2 \le N \le 750$) and A ($1 \le A \le 26$), which are the length of the string describing the chip and the number of letters in the alphabet respectively. The next line contains the string S, which is of length N and from an alphabet consisting of the first A letters of the alphabet. It is guaranteed that N is even.

The next A lines describe the letter pair value matrix V. Each line contains A integers each in the inclusive range from 1 to 1 000. The jth integer on the ith line is the value of a pairing between the ith letter of the alphabet and the jth letter of the alphabet.

Output

Display $N \times A$ lines. These lines can be divided into blocks of A. The jth line in the ith block is the value of the chip if the ith index in the string were modified to the jth letter of the alphabet. Since these values can be large, display each of them modulo $10^9 + 7$.

Sample Input 1

```
4 2
ABAB
1 2
3 4
```

Sample Output 1

```
10
20
4
10
10
16
5
10
```

Sample Input 2

```
6 3
ABCCAA
1 2 3
4 4 5
1 1 1
```

Sample Output 2

```
28
69
16
15
28
11
36
54
28
18
44
28
28
79
29
28
54
80
```

Sample Input 3

```
2 2
AB
1 2
3 4
```

Sample Output 3

```
2
4
1
2
```

Problem H
Hiring an Accountant
Time limit: 7 seconds

Jay's son ja$on is looking for an accountant to help him with his finances after selling OTM call options on NVDA. He has a list of N accountants which he interviews one by one, in order. After intreviewing the ith accountant he either offers the accountant a job or not. If he offers the accountant a job, the accountant will either accept or reject the offer. If the accountant accepts the offer, Jay's son ja$on will hire the accountant and will stop interviewing accountants. If the accountant rejects the offer, Jay's son ja$on will move on. However, he will miss the next M_i accountants and not be able to interview them due to wasted time on the rejected offer.

Each accountant has a skill level S_i and a probability C_i of accepting the offer. Jay's son ja$on wants to maximise the expected skill level of the accountant he hires because he minimised the value of his brokerage account.

Jay's son ja$on has asked you to come up with a strategy to maximise the expected skill level of the accountant he hires. The strategy you provide should tell him, for each accountant on the list to be interviewed, whether or not to offer them a job if he has a chance to (that is if all earlier candidates were not offered jobs or rejected the offers). Note that this means there are 2^N unique strategies you could provide Jay's son ja$on with. The expected skill level hired for a strategy is the average over a very large number of trials (e.g., 10^{1000}). Each trial is an independent application of the strategy with the accountants making their weighted random choices.

If Jay's son ja$on doesn't get an accountant, he will have to do it himself with his skill of 0.

Jay's son ja$on doesn't want to have to change his name back to Jay's son jason. So, please help Jay's son ja$on maximise the expected skill level of his accountant!

Input

The first line of the input contains a single integer N ($1 \leq N \leq 200\,000$), denoting the number of accountants to consider.

The second line contains N integers $S_1, S_2, \ldots, S_N$ ($1 \leq S_i \leq 10^6$), each denoting the skill of the ith accountant.

The third line contains N floating point numbers $C_1, C_2, \ldots, C_N$ ($0.0 \leq C_i \leq 1.0$), each denoting the probability that the ith accountant will accept Jay's son ja$on's offer. Each floating point number has at most 6 digits after the decimal point.

The fourth line contains N integers $M_1, M_2, \ldots, M_N$ ($0 \leq M_i \leq N$), denoting how many accountants Jay's son ja$on will miss the opportunity to interview if the ith accountant rejects an offer. It is possible that $i + M_i + 1 > N$.

Output

Display the maximum achievable expected skill level of the hired accountant to within an absolute or relative error of 10^{-6}.

Sample Input 1

```
5
4 1 2 3 5
0.15 0.8 0.16 0.5 0.7
2 4 0 1 1
```

Sample Output 1

```
3.575000000000000000
```

Sample Input 2

```
1
5
0.7
0
```

Sample Output 2

```
3.500000000000000000
```

Sample Input 3

```
6
789117 255689 492196 856719 468976 688464
0.078680 0.547505 0.535948 0.158366 0.566772 0.926440
4 0 0 6 1 0
```

Sample Output 3

```
649724.589843571200000000
```

Problem I
Investigator of Crashes
Time limit: 3 seconds

You have been hired as an Investigator of Crashes in Programmable Calculators. The calculator software you are investigating allows the definition of fixed point functions. Each such function consists of statements of the form `+x`, `-x`, `*x`, or `/x`, where `x` denotes some non-negative integer, except for division, where `x` is positive. Each operator appears exactly once, in some order.

The function takes a non-negative integer as input, and repeatedly (in a potentially endless loop) applies the specified operations to it in order, by adding, subtracting, multiplying with, or dividing by the given integer. Division is implemented as integer division and rounds down.

The software supports non-negative integers of arbitrary size, but cannot handle negative integers—if any operation would result in a negative value, the calculator crashes. If no crash occurs, the function will either run forever or until a fixed point is reached (that is, when applying the sequence of operations does not change the value), at which point it will stop.

Thus, when a non-negative integer is input to the function, there are exactly three possible outcomes: (1) it stops at some fixed point, (2) it crashes due to a negative number, or (3) it runs forever.

Consider the function `-1 /2 +3 *1`. An input value of 8 produces the sequence $8 \mapsto 6 \mapsto 5 \mapsto 5$, while an input of 1 produces $1 \mapsto 3 \mapsto 4 \mapsto 4$. So both 8 and 1 stop at a fixed point. An input of 0 will cause a crash since the first operation is to subtract 1, resulting in a negative number. There are no inputs that cause this function to run forever. However, all inputs will run forever for the function `+10 *3 /1 -0`.

You are given a list of functions. For each function, find the smallest non-negative input that will make the function stop, or state that such a number does not exist.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10\,000$), which is the number of functions to consider.

The next N lines describe the functions. Each of these lines contains exactly 4 statements of the form `+x`, `-x`, `*x`, or `/y`, with $0 \leq x \leq 10^6$ and $1 \leq y \leq 10^6$. Each operator appears exactly once.

Output

For each function, display the smallest non-negative input that will make each function stop. If there is no such input, display `BAD CODE` instead.

Sample Input 1	Sample Output 1
3 -1 /2 +3 *1 *3 -5 /2 +1 -1 /1 +0 *3	1 3 BAD CODE

This page is intentionally left (almost) blank.

Problem J
Jay's Son Ja$on's Family Tree
Time limit: 2 seconds

Jay's son ja$on has moved on from socks and is looking to start a family. But, he has some very particular naming conventions. His son (and his sons' sons and so on) will be named using the following rules:

The son's first name will be their father's first name concatenated with the letters "SON". The son's middle name will be the concatenation of their father's first, middle, and last names. The son's last name will be the same as their father's last name.

For example, Jay's son ja$on's name is of the following form (but using uppercase letters):

- First: JAY'S
- Middle: SON
- Last: JA$ON

Jay's son ja$on's son (which we would call his first descendant) would have the following first, middle, and last names (but using uppercase letters):

- First: JAY'SSON
- Middle: JAY'SSONJA$ON
- Last: JA$ON

And Jay's son ja$on's grandson, which we call his second descendant, would have the following first, middle, and last names (but using uppercase letters):

- First: JAY'SSONSON
- Middle: JAY'SSONJAY'SSONJA$ONJA$ON
- Last: JA$ON

Jay's son ja$on has convinced his friends to use this naming convention as well! For example, his best friend BENNY FACTORS BENEFACTOR will give his future son the following first, middle, and last names:

- First: BENNYSON

- Middle: BENNYFACTORSBENEFACTOR
- Last: BENEFACTOR

Unfortunately, Jay's son ja$on won't live forever so he can't make sure that this naming convention is followed for all of his and his friends' descendants. But, he can come back as a ghost and check a birth certificate. Since the birth certificate will be very long he only has time to check one character of his or one of his friend's Kth descendant. If he finds the wrong letter in the birth certificate, he will haunt this descendant for the rest of their life by stealing socks from the washing machine.

Jay's son ja$on needs your help to figure out the Cth index character (starting from 0) of his or one of his friend's Kth descendant's full name, assuming the naming convention has been followed correctly. The Cth index character in the full name of the Kth descendant is the Cth index character of the concatenation of their first, middle, and last name, without any spaces between.

Input

The first line contains two integers. The first integer K ($1 \le K \le 10^9$), denotes the depth of the target name in the family tree. The second integer C ($0 \le C \le 5 \times 10^{12}$), denotes which character of the target name to compute. It is guaranteed that C will be less than the length of the target name at depth K.

The second line contains three space-separated non-empty strings of uppercase English letters, dollar signs ($), or single quotes ('), denoting the first, middle, and last names, respectively, of the name at the top of the family tree. The length of each string will be at most 10^5 characters.

Output

Display a single character which is the Cth index character of the concatenation of the first, middle, and last name at depth K in the family tree.

Sample Input 1
```
1 23
JAY'S SON JA$ON
```
Sample Output 1
```
$
```

Sample Input 2
```
2 3
CARLS SON CARLSON
```
Sample Output 2
```
L
```

Sample Input 3
```
3 8
BENNY FACTORS BENEFACTOR
```
Sample Output 3
```
S
```

Sample Input 4
```
5 25
XBLDE HN EPGO
```
Sample Output 4
```
S
```

Problem K
Killing Monsters
Time limit: 12 seconds

You are playing your favourite video game Boulder's Gait. Each playthrough of Boulder's Gait happens over several rounds. In each round, you select a single monster to battle. Each monster has a particular payout associated with it. The monsters you choose from and their corresponding payouts are the same in every round. You may battle the same monster multiple times in different rounds, and you will receive the same payout each time you battle them.

Your squad is very picky and only wants to do playthroughs where the total payout across all battles can be evenly divided amongst the members of the squad (call those playthroughs *fair*). For example, consider the following monsters and their payouts. Each of the playthroughs listed are fair if your squad has three members because the total payout can be evenly divided amongst the three members of the squad.

Boss A Payout: 2 💰 | **Boss B** Payout: 5 💰 | **Boss C** Payout: 3 💰

Playthrough 1: 6 💰	Playthrough 2: 3 💰	Playthrough 3: 9 💰	Playthrough 4: 6 💰
Round 1: A Round 2: A Round 3: A	Round 1: C	Round 1: B Round 2: A Round 3: A	Round 1: C Round 2: C

Two playthroughs are considered the same if they have the exact same number of rounds and the order in which you battle the monsters is identical.

In order for your playthrough to be eligible for the leaderboard, your playthrough must have at least L rounds and at most U rounds.

How many different fair playthroughs are there that are eligible for the leaderboard?

Input

The first line of input contains four integers N ($1 \leq N \leq 200\,000$), which is the number of monsters to choose from, K ($1 \leq K \leq 750$), which is the number of members in your squad, L ($1 \leq L \leq 10^9$), which is the minimum number of rounds for your playthrough to be eligible for the leaderboard, and U ($L \leq U \leq 10^9$), which is the maximum number of rounds for your playthrough to be eligible for the leaderboard.

The next line contains N integers $b_1, b_2, \ldots, b_N$ ($1 \leq b_i \leq 10^9$), which are the payouts of the N monsters.

Output

Display the number of different fair playthroughs that are eligible for the leaderboard. Display this number modulo $1\,000\,000\,007$.

Sample Input 1	Sample Output 1
`3 3 1 3` `2 5 3`	`11`

Sample Input 2	Sample Output 2
`2 1 5 6` `10 10`	`96`

Sample Input 3	Sample Output 3
`3 5 1 1` `15 7 10`	`2`

Problem L
Lonely Wizard
Time limit: 5 seconds

Nathan is a thirty-year-old wizard who lives in a small town. He is very lonely and awkward and wants to use his wizard powers to make everyone else lonely and awkward.

He has a list of N people in the town, and he knows which people are friends with each other. Friendships are bidirectional: if A is friends with B, then B is friends with A. He will use his wizard powers to partition the N people into two disjoint (possibly empty) sets. Everyone must be in exactly one of the two sets. People who are in different sets cannot communicate and will stop being friends as a result. People in the same set who are already friends will stay friends; there is nothing Nathan can do about this. Note that no new friendships will be created.

Nathan wants to cause the most division and strife possible, so he wants to maximise the number of friendships that he destroys when partitioning the people into two sets. Help Nathan find the maximum number of friendships that he can destroy by partitioning the people into two sets.

Input

The first line of the input contains two integers N $(1 \leq N \leq 16)$, which is the number of people in the town, and F $(0 \leq F \leq \frac{N \times (N-1)}{2})$, which is the number of friendships.

The following F lines each contain two integers A, B $(1 \leq A < B \leq N)$, denoting a friendship between the persons A and B. No friendship is duplicated in the input.

Output

Display a single integer, which is the maximum number of friendships that Nathan can destroy.

Sample Input 1	Sample Output 1
`1 0`	`0`

Sample Input 2	Sample Output 2
`2 1` `1 2`	`1`

Sample Input 3	Sample Output 3
4 4 1 2 1 4 1 3 2 4	3