Problem A. Magic

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke participated in a magic show.

A magician prepared N identical-looking boxes. He put a treasure in one of the boxes, closed the boxes, shuffled them, and numbered them 1 through N.

Since the boxes are shuffled, now Snuke has no idea which box contains the treasure. Snuke wins the game if he opens a box containing the treasure. You may think that if Snuke opens all boxes at once, he can always win the game. However, there are some tricks:

- Snuke must open the boxes one by one. After he opens a box and checks the content of the box, he must close the box before opening the next box.
- He is only allowed to open Box i at most a_i times.
- The magician may secretly move the treasure from a closed box to another closed box, using some magic trick. For simplicity, assume that the magician never moves the treasure while Snuke is opening some box. However, he can move it at any other time (possibly before Snuke opens the first box, or between he closes some box and opens the next box).
- The magician can perform the magic trick at most K times.

Can Snuke always win the game, regardless of the initial position of the treasure and the movements of the magician?

Input

Input is given in the following format:

- Line 1: N K
- Line 2: a_1 a_2 $\cdots$ a_N

Constraints:

$2 \leq N \leq 50$, $1 \leq K \leq 50$ and $1 \leq a_i \leq 100$

Output

If the answer is no, print a single -1.

Otherwise, print one possible move of Snuke in the following format:

- Line 1: Q
- Line 2: x_1 x_2 $\cdots$ x_Q

It means that he opens boxes $x_1, x_2, \cdots, x_Q$ in this order.

In case there are multiple possible solutions, you can output any.

Examples

standard input	standard output
2 1 5 5	7 1 1 2 1 2 2 1
3 50 5 10 15	-1

Note

In Sample 1, if Snuke opens the boxes 7 times in the order $1 \to 1 \to 2 \to 1 \to 2 \to 2 \to 1$, he can always find the treasure regardless of its initial position and the movements of the magician.

Problem B. Multiple of Nine

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

Count the number of strings S that satisfy the following constraints, modulo $10^9 + 7$.

- The length of S is exactly N.
- S consists of digits (‘0‘...‘9‘).
- You are given Q intervals.

For each i $(1 \le i \le Q)$, the integer represented by $S[l_i \dots r_i]$ (the substring of S between the l_i-th (1-based) character and the r_i-th character, inclusive) must be a multiple of 9.

Here, the string S and its substrings may have leading zeroes. For example, “002019” represents the integer 2019.

Input

Input is given in the following format:

- Line 1: N
- Line 2: Q
- Line 3: l_1 r_1
- ...
- Line $Q + 2$: l_Q r_Q

Constraints:

$1 \le N \le 10^9$, $1 \le Q \le 15$, $1 \le l_i \le r_i \le N$.

Output

Print the number of strings that satisfy the conditions, modulo $10^9 + 7$.

Examples

standard input	standard output
4 2 1 2 2 4	136
6 3 2 5 3 5 1 3	2720
20 10 2 15 5 6 1 12 7 9 2 17 5 15 2 4 16 17 2 12 8 17	862268030

Note

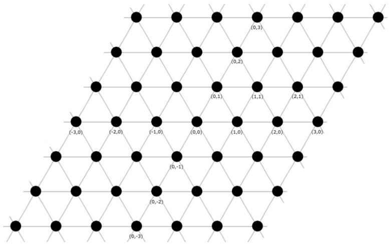
:

In Sample 1, S = “9072” satisfies the conditions because both $S[1 \dots 2]$ =“90” and $S[2 \dots 4]$ = “072” represent multiples of 9.

Problem C. Triangular Lamps

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

There is an infinitely large triangular grid, as shown below. Each point with integer coordinates contains a lamp.



Initially, only the lamp at ¡font color=”red”¿**(X, Y)**¡/font¿ was on, and all other lamps were off. Then, Snuke performed the following operation zero or more times:

Choose two integers x and y. Toggle (on to off, off to on) the following three lamps: $(x, y), (x, y + 1), (x + 1, y)$.

After the operations, N lamps $(x_1, y_1), \cdots, (x_N, y_N)$are on, and all other lamps are off. Find X and Y.

Input

Input have the following format:

- Line 1: N
- Line 2: x_1 y_1
- ...
- Line $N + 1$: x_N y_N

Constraints:

$1 \le N \le 10^4$, $-10^{17} \le x_i, y_i \le 10^{17}$, (x_i, y_i) are pairwise distinct; the input is consistent with the statement, and you can uniquely determine X and Y.

Output

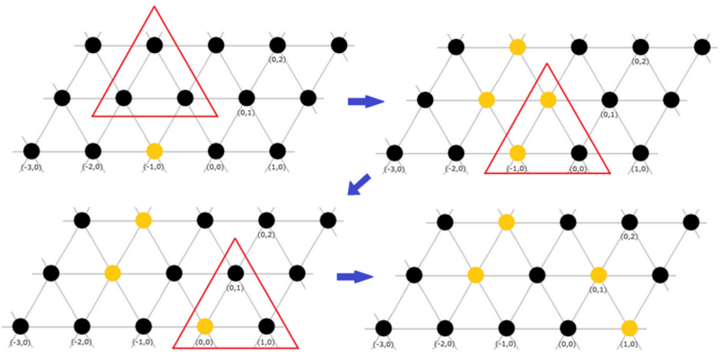
Print X and Y, separated by a space.

Example

standard input	standard output
4 -2 1 -2 2 0 1 1 0	-1 0

Note

The following picture shows one possible sequence of operations for the sample:



Problem D. Distinct Boxes

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Snuke has R red balls and B blue balls. He distributes them into K boxes, such that no box is empty and no two boxes are identical. Compute the maximum possible value of K.

Formally speaking, let's number the boxes 1 through K. If Box i contains r_i red balls and b_i blue balls, the following conditions must be satisfied:

- For each i $(1 \le i \le K)$, $r_i > 0$ or $b_i > 0$.
- For each i, j $(1 \le i < j \le K)$, $r_i \ne r_j$ or $b_i \ne b_j$.
- $\sum r_i = R$ and $\sum b_i = B$ (no balls can be left outside the boxes).

Input

Input is given in the following format:

- Line 1: R B

Constraints:

$1 \le R, B \le 10^9$

Output

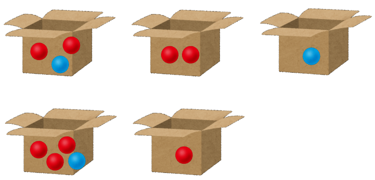
Print the maximum possible value of K.

Example

standard input	standard output
8 3	5

Note

The following picture shows one possible way to achieve $K = 5$:



Problem E. e

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

There is a very long bench. The bench is divided into M sections, where M is a very large integer.

Initially, the bench is vacant. Then, M people come to the bench one by one, and perform the following action:

We call a section *comfortable* if the section is currently unoccupied and is not adjacent to any occupied sections. If there is no comfortable section, the person leaves the bench. Otherwise, the person chooses one of comfortable sections uniformly at random, and sits there. (The choices are independent from each other).

After all M people perform actions, Snuke chooses an interval of N consecutive sections uniformly at random (from $M - N + 1$ possible intervals), and takes a photo. His photo can be described by a string of length N consisting of '`X`' and '`-`': the i-th character of the string is '`X`' if the i-th section from the left in the interval is occupied, and '`-`' otherwise. Note that the photo is directed. For example, "`-X--X`" and "`X--X-`" are different photos.

What is the probability that the photo matches a given string s? This probability depends on M. You need to compute the limit of this probability when M goes infinity.

Here, we can prove that the limit can be uniquely written in the following format using three *rational* numbers p, q, r and $e = 2.718\ldots$ (the base of natural logarithm):

$$p + \frac{q}{e} + \frac{r}{e^2}$$

Your task is to compute these three rational numbers, and print them modulo $10^9 + 7$, as described in Output format.

Input

Input is given in the following format:

- Line 1: N
- Line 2: s

Constraints:

- $1 \le N \le 1000$
- $|s| = N$
- s consists of '`X` and '`-`'.

Output

Print three rational numbers p, q, r, separated by spaces. When you print a rational number, first write it as a fraction $\frac{y}{x}$, where x, y are integers and x is not divisible by $10^9 + 7$ (under the constraints of the problem, such representation is always possible). Then, you need to print the only integer z between 0 and $10^9 + 6$, inclusive, that satisfies $xz \equiv y \pmod{10^9 + 7}$.

Examples

standard input	standard output
1 X	500000004 0 500000003
3 ---	0 0 0
5 X--X-	0 0 1
5 X-X-X	500000004 0 833333337
20 -X--X--X-X--X--X-X-X	0 0 183703705

Note

For Sample 1, the probability that a randomly chosen section is occupied converge to $\frac{1}{2} - \frac{1}{2e^2}$.

For Sample 2, after the actions, no three consecutive unoccupied sections can be left.

For Sample 3, the limit is $\frac{1}{e^2}$.

For Sample 4, the limit is $\frac{1}{2} - \frac{13}{6e^2}$.

For Sample 5, the limit is $\frac{7}{675e^2}$.

The `samples.zip` file contains six samples; in sixth sample $N = 100$.

Problem F. Paths

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

You are given a simple graph with N vertices and M edges. The vertices are numbered from 1 through N, and the edges are numbered 1 through M. The edge i connects vertices A_i and B_i.

For each vertex x ($1 \le x \le N$), count the number of 4-tuples of vertices (v_1, v_2, v_3, v_4) that satisfy the following conditions: vertices x, v_1, v_2, v_3, v_4 are distinct from each other, the graph has edges (x, v_1), (v_1, v_2), (v_2, v_3), and (v_3, v_4).

Input

First line of the input contains two integers N and M ($1 \le N, M \le 10^5$). Each of next M lines contain pair of vertices A_i and B_i connected by the edge ($1 \le A_i, B_i \le N$). It is guaranteed that the given graph is simple.

Output

Print N lines. In the i-th line, print the number of 4-tuples of vertices (v_1, v_2, v_3, v_4) that satisfy the abovementioned conditions for the vertex i.

Examples

standard input	standard output
6 8 1 2 2 3 5 2 1 5 3 4 5 3 5 6 5 4	7 4 4 7 2 4
6 15 1 2 1 3 1 4 1 5 1 6 2 3 2 4 2 5 2 6 3 4 3 5 3 6 4 5 4 6 5 6	120 120 120 120 120 120
8 11 3 6 7 8 8 5 4 5 2 1 2 6 3 2 4 3 5 3 6 5 7 6	12 14 11 17 11 9 15 17

Problem G. Ranges

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

You have a (multi)set S of ranges. You have to process N queries. The i-th query is one of the following types:

- $T_i = 1$: You are given two integers L_i and R_i. Add the range $[L_i, R_i]$ to S.
- $T_i = 2$: You are given an integer K_i. remove the K_i-th added range from S. Note that counting starts from 0. For example, the first added range is the 0-th added range.
- $T_i = 3$: You are given two integers L_i and R_i. For each range $[l, r] \in S$ that satisfies $l < L_i$ and $R_i < r$, calculate $(L_i - l) \times (r - R_i)$, and find the maximum value of them. If no such range exists, answer -1.

Input

Input is given from Standard Input in the following format:

- line 1: N
- line $1 + i$: $T_i \cdots$

The line $1 + i$ contains the information of the i-th query. If $T_i = 1$, it is followed by two integers L_i and R_i. If $T_i = 2$, it is followed by an integer K_i. If $T_i = 3$, it is followed by two integers L_i and R_i.

Constraints:

$1 \le N \le 2 \times 10^5$, $1 \le T_i \le 3$, $0 \le L_i \le R_i \le 10^9$. It is guaranteed that the K_i-th added range exists in S at the moment of the i-th query. All values in input are integers.

Output

For each query with $T_i = 3$, output the answer to the query in a line.

Examples

standard input	standard output
9 1 0 10 1 2 12 3 5 5 3 8 9 2 0 3 5 5 3 8 9 2 1 3 5 5	25 18 21 18 -1
7 1 3 7 1 0 6 1 4 10 3 1 3 3 6 7 3 5 5 3 4 6	3 6 5 1
20 3 268323303 605806817 3 397106901 526645597 1 242167025 963419065 3 306157656 666722488 3 90905255 723611215 1 135062270 656996756 1 72048580 708895403 1 254360876 741288738 3 353173849 652094091 3 274378199 520888695 1 128877839 722596185 3 367349293 905356554 3 336742409 649201453 1 353239854 521572577 2 3 3 5185803 799351855 1 139746807 783110900 3 375190636 656724546 1 462675641 992773167 1 77055484 555060299	-1 -1 18985801177770087 -1 34559196595622576 38039325599084252 7268396812754948 29717251314463008 -1 40797612391288109

Problem H. Simple Game

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

Rng and Snuke are going to play a simple game. Initially, Rng has X candies in his box and Snuke has Y candies in his box. The game consists of N turns. Starting from Rng, they take turns alternately. In the i-th turn, the player takes A_i candies from the opponent's box and put them in his own box. If the opponent's box contains less than A_i candies, the player takes all of them.

Since the game is too simple, they are going to process Q queries to change a parameter of the game. The i-th query is one of the following types:

- $T_i = 1$: An integer V_i is given. Change X to V_i.
- $T_i = 2$: An integer V_i is given. Change Y to V_i.
- $T_i = 3$: Two integers S_i and V_i are given. Change A_{S_i} to V_i.

After each query, find the number of candies that Rng has at the end of the game if they play the game with current parameters.

Input

Input is given in the following format:

- line 1: N
- line 2: X Y
- line 3: A_1 A_2 $\cdots$ A_N
- line 4: Q
- line $4 + i$: T_i $\cdots$

The line $4 + i$ contains the information of the i-th query. If $T_i = 1$, it is followed by an integer V_i. If $T_i = 2$, it is followed by an integer V_i. If $T_i = 3$, it is followed by two integers S_i and V_i.

Constraints:

- $1 \le N \le 2 \times 10^5$
- $1 \le X \le 10^9$
- $1 \le Y \le 10^9$
- $1 \le A_i \le 10^9$
- $1 \le Q \le 2 \times 10^5$
- $1 \le T_i \le 3$
- $1 \le S_i \le N$
- $1 \le V_i \le 10^9$
- All values in input are integers.

Output

Print Q lines. In the i-th line, print the number of candies that Rng has at the end of the game if Rng and Snuke play the game right after the i-th query.

Examples

standard input	standard output
3 5 4 2 1 4 3 2 7 3 2 3 2 2	10 8 7
5 11 21 10 5 12 1 13 5 3 3 1 2 8 2 17 3 4 10 3 4 25	29 19 28 20 13
5 11 4 13 6 7 3 5 4 3 4 10 3 2 11 3 4 18 3 3 2	10 6 5 5
4 4 18 2 9 19 6 5 1 7 2 8 1 1 2 4 1 9	13 9 3 0 7

Problem I. Union 2SAT

Input file:	*standard input*
Output file:	*standard output*
Time limit:	6 seconds
Memory limit:	512 mebibytes

There are N boolean variables $X_1, X_2, \cdots, X_N$.

There are M slimes numbered from 1 through M. The slime i is satisfied if at least one of the following conditions are met.

- $X_{|A_i|} = (A_i > 0)$
- $X_{|B_i|} = (B_i > 0)$

You are going to merge slimes $M-1$ times. In the i-th merge operation, you merge slimes P_i and Q_i, and get a new slime numbered $M+i$. After merging, slimes P_i and Q_i disappear. The new slime is satisfied when both slimes P_i and Q_i are satisfied.

For each $i(M+1 \leq i \leq M+M-1)$, determine wether it is possible to assign values to $X_1, X_2, \cdots, X_N$ so that the slime i is satisfied.

Input

Input is given from Standard Input in the following format:

- line 1: N M
- line $1+i$: A_i B_i
- line $1+M+i$: P_i Q_i

Constraints:

- $2 \leq N \leq 10^5$
- $2 \leq M \leq 10^5$
- $1 \leq |A_i| \leq N$
- $1 \leq |B_i| \leq N$
- $|A_i| \neq |B_i|$
- $(A_i, B_i) \neq (A_j, B_j)$ and $(A_i, B_i) \neq (B_j, A_j)$ $(i \neq j)$
- $1 \leq P_i < M+i$
- $1 \leq Q_i < M+i$
- $P_i \neq Q_i$
- It is guaranteed that slimes P_i and Q_i exist at the moment of the i-th merge operation.

Output

Print $M-1$ lines. In the i-th line, print "`Possible`" if it is possible to satisfy the slime $M+i$, and print "`Impossible`" otherwise.

Examples

standard input	standard output
3 4 1 2 -2 -3 -1 3 1 -2 1 3 2 5 6 4	Possible Possible Possible
3 4 -1 -2 2 3 1 -2 2 -3 1 4 2 5 6 3	Possible Possible Impossible
4 20 3 -2 4 -3 3 -1 -2 -3 -1 2 2 -4 2 -3 4 -1 1 4 -1 -3 4 2 1 -3 -2 -1 2 3 3 -4 -4 -2 -4 1 -1 -4 2 1 -4 -3 1 11 16 2 6 20 8 12 22 15 18 5 23 7 9 10 13 26 19 21 4 3 28 31 14 24 27 30 29 25 17 34 36 32 35 33 37 38	Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Possible Impossible Possible Impossible