Assigement 2

Deadline 31/1/2021

Max Number of students is **4**

Solve using any programming language.
Deliver .cpp Only on BlackBoard

Problem 1

Assume that a disk scheduler is to handle file data requests A, B, and C to tracks (or cylinders) 100, 50, and 190, respectively, and that the disk read/write head is initially at track 140. Assume it takes 1 time-unit for the disk to seek (move) a distance of one track. If the file data requests are handled in first-come-first-serve order (A,B,C), then the total seek time is 40+50+140=230 time-units. Suppose the requests S are to be serviced in the order which minimizes total seek time instead. The goal is to compute the minimal seek time f({A,B,C}, 140) = 190, which can be achieved by the order (C,A,B).

Problem 2

You are so lucky in 2021 and finally you have bought your dream car. You decided to travel to many cities. The car tank capacity is **C** litres. And each trip will cost x liters of fuel. You can fill the tank when visiting each city.

The expected inputs

N : number of cities
C : Tank capacity
A[]: the fuel amount you can fill from each city
B[]:the fuel consumption to travel from city i and city i+1

The output
The maximum number of cities you can travel

Problem 3

You are given a connected weighted undirected graph without any loops and multiple edges. Let us remind you that a graph's spanning tree is defined as an acyclic connected sub graph of the given graph that includes all of the graph's vertexes. The weight of a tree is defined as the sum of weights of the edges that the given tree contains. The minimum spanning tree (**MST**) of a graph is defined as the graph's spanning tree having the minimum possible weight. For any connected graph obviously exists the minimum spanning tree, but in the general case, a graph's minimum spanning tree is not unique.

Your task is to determine the following for each edge of the given graph: whether it is either included in **any** MST, or included **at least in one** MST, or **not included in any** MST.

Input

The first line contains two integers n and m ($2 \le n \le 10^5$, $n - 1 \le m \le min(10^5, \frac{n(n-1)}{2})$) — the number of the graph's vertexes and edges, correspondingly. Then follow m lines, each of them contains three integers — the description of the graph's edges as "a_i b_i w_i" ($1 \le a_i, b_i \le n$, $1 \le w_i \le 10^6$, $a_i \ne b_i$), where a_i and b_i are the numbers of vertexes connected by the i-th edge, w_i is the edge's weight. It is guaranteed that the graph is connected and doesn't contain loops or multiple edges.

Output

Print m lines — the answers for all edges. If the i-th edge is included in any MST, print "`any`"; if the i-th edge is included at least in one MST, print "`at least one`"; if the i-th edge isn't included in any MST, print "`none`". Print the answers for the edges in the order in which the edges are specified in the input.

Examples

Input#1

```
4 5
1 2 101
1 3 100
2 3 2
2 4 2
3 4 1
```

Output#1

```
none
any
at least one
at least one
any
```

input#2

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

Output#2

```
any
any
none
```

input#3

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

Output#3

```
at least one
at least one
at least one
```

Note

In the second sample the MST is unique for the given graph: it contains two first edges.

In the third sample any two edges form the MST for the given graph. That means that each edge is included at least in one MST.

Problem 4

Ilya is working for the company that constructs robots. Ilya writes programs for entertainment robots, and his current project is "Bob", a new-generation game robot. Ilya's boss wants to know his progress so far. Especially he is interested if Bob is better at playing different games than the previous model, "Alice".

So now Ilya wants to compare his robots' performance in a simple game called "1-2-3". This game is similar to the "Rock-Paper-Scissors" game: both robots secretly choose a number from the set $\{1, 2, 3\}$ and say it at the same moment. If both robots choose the same number, then it's a draw and noone gets any points. But if chosen numbers are different, then one of the robots gets a point: 3 beats 2, 2 beats 1 and 1 beats 3.

Both robots' programs make them choose their numbers in such a way that their choice in $(i + 1)$-th game depends only on the numbers chosen by them in i-th game.

Ilya knows that the robots will play k games, Alice will choose number a in the first game, and Bob will choose b in the first game. He also knows both robots' programs and can tell what each robot will choose depending on their choices in previous game. Ilya doesn't want to wait until robots play all k games, so he asks you to predict the number of points they will have after the final game.

Input

The first line contains three numbers k, a, b ($1 \le k \le 10^{18}$, $1 \le a, b \le 3$).

Then 3 lines follow, i-th of them containing 3 numbers $A_{i,1}$, $A_{i,2}$, $A_{i,3}$, where $A_{i,j}$ represents Alice's choice in the game if Alice chose i in previous game and Bob chose j ($1 \le A_{i,j} \le 3$).

Then 3 lines follow, i-th of them containing 3 numbers $B_{i,1}$, $B_{i,2}$, $B_{i,3}$, where $B_{i,j}$ represents Bob's choice in the game if Alice chose i in previous game and Bob chose j ($1 \le B_{i,j} \le 3$).

Output

Print two numbers. First of them has to be equal to the number of points Alice will have, and second of them must be Bob's score after k games.

Examples

Input#1

```
10 2 1
1 1 1
1 1 1
1 1 1
2 2 2
2 2 2
2 2 2
```

Output#1

```
1 9
```

Input#2

```
8 1 1
2 2 1
3 3 1
3 1 3
1 1 1
2 1 1
1 2 3
```

Output#2

```
5 2
```

Input#3

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

Output#3

```
0 0
```

Note

In the second example game goes like this:

$$(1,1) \to (2,1) \to (3,2) \to (1,2) \to (2,1) \to (3,2) \to (1,2) \to (2,1)$$

The fourth and the seventh game are won by Bob, the first game is draw and the rest are won by Alice.