Kick Start 2022 - Round E

Pizza Delivery

Problem

Ada delivers pizzas in a city consisting of a grid of $\mathbf{N}$ horizontal and $\mathbf{N}$ vertical streets, heading from North to South and from West to East, respectively, and numbered from 1 to $\mathbf{N}$. The top left street crossing of the grid is $(1, 1)$.

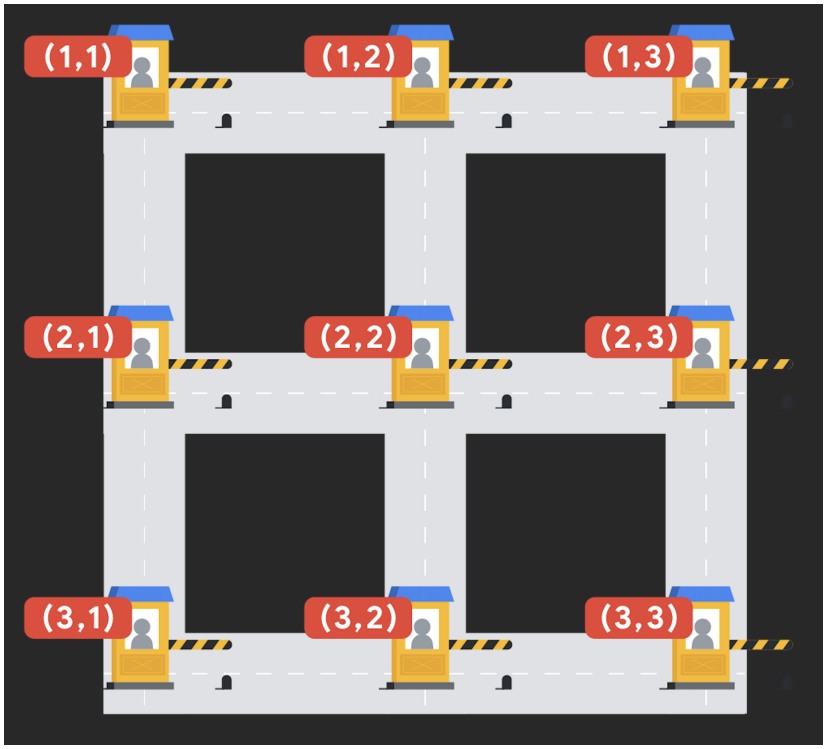
Today, Ada has to deliver $\mathbf{P}$ pizzas, one for each of $\mathbf{P}$ customers. Each customer lives at a different street crossing; the k-th customer lives at street crossing $(\mathbf{X_k}, \mathbf{Y_k})$ and will pay Ada $\mathbf{C_k}$ coins after the pizza is delivered to their location.

Ada starts at her pizza restaurant at $(\mathbf{A_r}, \mathbf{A_c})$ with 0 coins and carrying $\mathbf{P}$ pizzas. Her goal is to deliver all of the pizzas within $\mathbf{M}$ minutes. She is free to take any path she likes around the city and finish deliveries anywhere, as long as she manages to drop off all $\mathbf{P}$ pizzas to their respective customers within $\mathbf{M}$ minutes. It takes one minute to walk between two adjacent street crossings, and takes no additional time to drop off a pizza at a customer's location. There are some additional rules and constraints to note:

- Ada is not allowed to go outside the grid.
- No customer lives at the same street crossing as the pizza restaurant Ada starts her trip.
- At any point in time Ada can choose to stay in her current location and not move.
- Ada can also choose not to deliver a pizza when at a customer's location.

Formally, if Ada is currently at street crossing (i, j), where i is i-th row from the top, and j is j-th column from the left, she can decide to do any of the following as long as she does not go outside the grid:

- Go north, she reaches street crossing $(i - 1, j)$.
- Go east, she reaches street crossing $(i, j + 1)$.
- Go west, she reaches street crossing $(i, j - 1)$.
- Go south, she reaches street crossing $(i + 1, j)$.
- Stay at street crossing (i, j).



The city has a unique toll system in place for using the streets. There is a toll for using each street and the toll depends on Ada's current number of coins and the direction in which she is travelling to. The toll function is defined for every cardinal direction (North, East, West, South) separately. The toll function F_d for $d \in \{North, East, West, South\}$ returns the amount of coins Ada will have after moving in the direction d and is defined as follows:

$$F_d = c \ \mathbf{OP_d} \ \mathbf{K_d}$$

where c is the current number of coins that Ada has and $\mathbf{OP_d}$ is an operator and $\mathbf{K_d}$ is a fixed positive integer. The allowed operators are:

- $+$ (addition),
- $-$ (subtraction),
- $*$ (multiplication),
- $/$ (integer division).

For example, we can have $F_{North} = c + 3$, $F_{East} = c * 4$, $F_{West} = c - 4$, $F_{South} = c / 2$. That means that if Ada moves North one street then she will have 3 more coins; if Ada moves East then Ada's coins will quadruple; if Ada moves West then she loses 4 coins; and if Ada moves South then her coins are halved.

All divisions are integer divisions and are computed by using floor function. For example, $\frac{-1}{4} = \lfloor \frac{-1}{4} \rfloor = -1$. Notice that Ada is allowed to have a negative number of coins. Note that the tolls might actually give Ada coins.

Find out if Ada can deliver all the $\mathbf{P}$ pizzas in $\mathbf{M}$ minutes and, if so, the maximum number of coins Ada could have after $\mathbf{M}$ minutes.

Input

The first line of the input gives the number of test cases, $\mathbf{T}$. $\mathbf{T}$ test cases follow.

The first line of each test case contains $\mathbf{N}$, $\mathbf{P}$, $\mathbf{M}$, $\mathbf{A_r}$, $\mathbf{A_c}$ denoting the grid size, the number of pizzas to deliver, the minutes in which all pizzas should be delivered, and the coordinates of the street crossing at which Ada starts respectively.

The next four lines denote the toll functions for North, East, West, South respectively. Each of these lines contains $\mathbf{OP_d}$, denoting the operator (one of `+`, `-`, `*`, `/`), and $\mathbf{K_d}$, the positive integer used in toll function.

The following $\mathbf{P}$ lines describe the customers. Each of these lines consists of three integers $\mathbf{X_k}$, $\mathbf{Y_k}$, $\mathbf{C_k}$ denoting the row number of the k-th customer from the top of the grid, the column number of the k-th customer from the left of the grid, and the amount of coins they pay on delivery, respectively.

Output

For each test case, output one line containing `Case #`x`:` y, where x is the test case number (starting from 1) and y is `IMPOSSIBLE` if all pizzas cannot be delivered within $\mathbf{M}$ minutes; otherwise, the output should be the maximum number of coins Ada can have after $\mathbf{M}$ minutes (which could be negative).

Limits

Time limit: 20 seconds.
Memory limit: 1 GB.
$1 \leq \mathbf{T} \leq 100$.
$1 \leq \mathbf{N} \leq 10$.
$1 \leq \mathbf{M} \leq 20$.
$1 \leq \mathbf{A_r}, \mathbf{A_c} \leq \mathbf{N}$.
$1 \leq \mathbf{X_k}, \mathbf{Y_k} \leq \mathbf{N}$, for all k.
$1 \leq \mathbf{C_k} \leq 4$, for all k.
$1 \leq \mathbf{K_d} \leq 4$, for all d.
$\mathbf{OP_d}$ is one of (`+`, `-`, `*`, `/`), for all d.
It is guaranteed that no customer lives at the same street crossing as the pizza restaurant Ada starts her trip, i.e. $(\mathbf{X_k}, \mathbf{Y_k}) \neq (\mathbf{A_r}, \mathbf{A_c})$, for all $1 \leq k \leq \mathbf{P}$.
It is guaranteed that every customer lives at a different street crossing, i.e. $(`X_k, \mathbf{Y_k}) \neq (\mathbf{X_l}, \mathbf{Y_l})$, for all $1 \leq k, l \leq \mathbf{P}$ and $k \neq l$.

Test Set 1

$\mathbf{P} = 0$.

Test Set 2

$0 \leq \mathbf{P} \leq 10$.

Sample

Note: there are additional samples that are not run on submissions down below.

Sample Input

```
2
3 0 1 1 2
+ 1
- 2
+ 3
/ 4
3 0 1 2 3
```

Sample Output

```
Case #1: 3
Case #2: 0
```

```
- 2
- 2
- 2
- 2
```

In Sample Case #1, Ada does not deliver any pizzas. Ada is located at street crossing $(1, 2)$. In 1 minute she can decide to move west and go to $(1, 1)$, so that the toll at $(1, 2)$ calculates Ada's coins using the toll function `F_{West} = c + 3`, which results in 3 coins. Therefore Ada can have maximum of 3 coins with her after 1 minute.

In Sample Case #2, Ada does not deliver any pizzas. Ada is located at street crossing $(2, 3)$. All directions have a similar toll function, $F_d = c - 2$ for all d. If she decides to go in any direction, she will end up with -2 coins. It is optimal for Ada to stay at the same location and have 0 coins at the end.

Additional Sample - Test Set 2

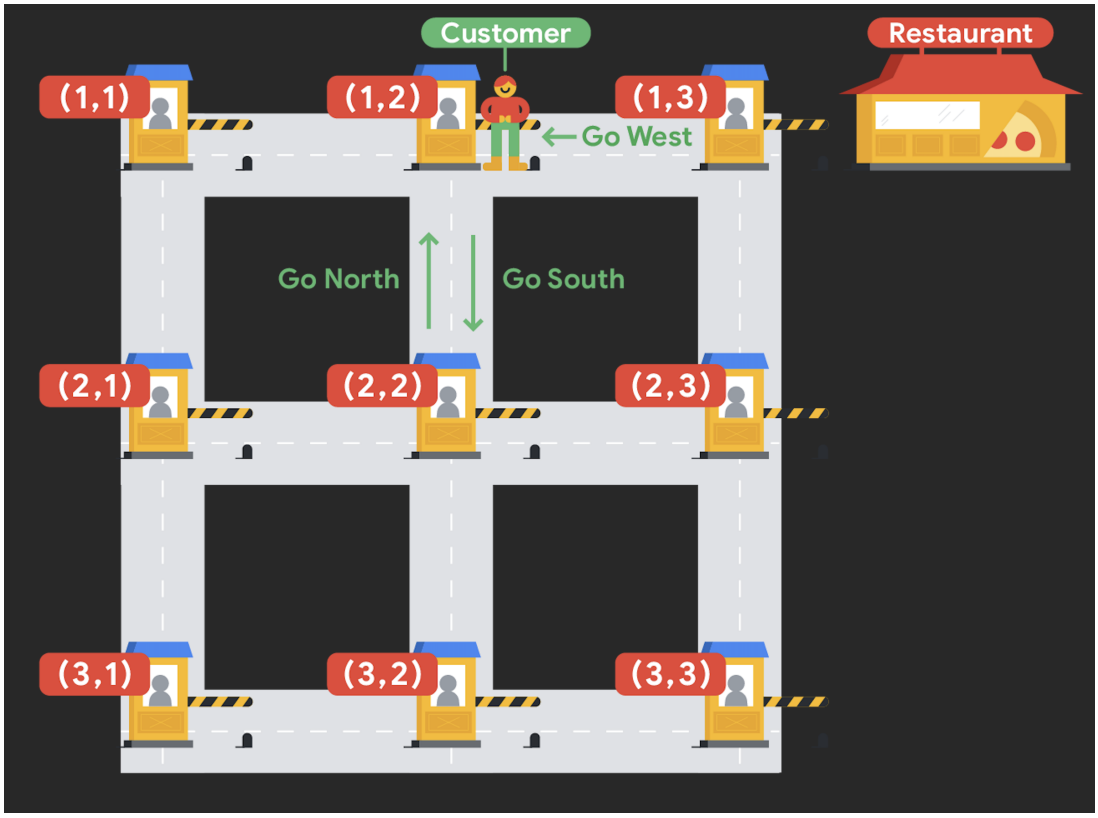
The following additional sample fits the limits of Test Set 2. It will not be run against your submitted solutions.

Sample Input

```
3
3 1 3 1 3
+ 4
- 2
* 1
/ 4
1 2 4
2 2 1 1 2
+ 2
+ 3
* 2
* 1
1 1 4
2 2 1
3 1 2 1 3
+ 1
* 1
- 3
/ 4
2 2 2
```

Sample Output

```
Case #1: 8
Case #2: IMPOSSIBLE
Case #3: 1
```



In Additional Sample Case #1, Ada started at street crossing $(1, 3)$ with 0 coins. Ada can receive maximum coins by following the steps below:

- Go west to $(1, 2)$. Using the toll function for moving west $F_{west} = c * 1$, Ada now has $0 * 1 = 0$ coins.
- Do not deliver the pizza at $(1, 2)$ yet, and go south to $(2, 2)$. Using the toll function for moving south $F_{South} = c / 4$, Ada now has $0/4 = 0$ coins.
- Go north to $(1, 2)$. Using the toll function for moving north $F_{North} = c + 4$, Ada now has $0 + 4 = 4$ coins.
- Deliver the pizza at $(1, 2)$. Ada receives 4 additional coins for delivering the pizza. Ada has a total of 8 coins in the end.

In Additional Sample Case #2, Ada cannot deliver two pizzas in one minute, so the output is `IMPOSSIBLE`.

In Additional Sample Case #3, Ada started at street crossing $(1, 3)$ with 0 coins. Ada can receive maximum coins by following the steps below:

- Go west to $(1, 2)$. Using the toll function for moving west $F_{West} = c - 3$, Ada now has $0 - 3 = -3$ coins.
- Go south to $(2, 2)$. Using the toll function for moving south $F_{South} = c / 4$, Ada now has $-3/4 = \lfloor \frac{-3}{4} \rfloor = -1$ coins.
- Deliver the pizza at $(2, 2)$. Ada receives 2 additional coins for delivering the pizza. Ada has a total of 1 coins in the end.