TimBot Specification

This is a robot simulation where robots compete for supremacy of the Doh-Nat Planet.

Background

TimBots live on Doh-Nat. A torus-shaped world that has limited resources. TimBots need power to survive, but the only power source on Doh-Nat is Spresso, a plant that is neither plentiful, nor fast-growing. Hence, the TimBots have concluded that eliminating each other is the best way to ensure their own survival.

The surface of Doh-nat is an $m \times n$ grid, where the right edge of the grid is adjacent to the left edge, and the top edge is adjacent to the bottom edge. Each square in the grid is called a district, and grows a limited number of Spresso plants. One district-worth of Spresso plants provides a TimBot with a few Jolts of energy, which a TimBot needs to function. Once the Spresso has been harvested from a district, it takes several days to grow back. To survive, the TimBots must roam the planet, feeding on Spresso, and eliminating their competitors in the process.

A TimBot can move from one district to another, harvest Spresso, fire its ion cannon at its competitors, and defend itself with a force-shield. Each of actions costs the TimBot one Jolt of energy. If a TimBot runs out of energy, it becomes nonfunctional.

Rules of the Simulation

The simulation takes place on an $m \times n$ torus grid, where the left edge is adjacent to the right edge, and the top edge is adjacent to bottom edge. I.e., it is impossible to fall off the grid—moving across one edge of the grid moves the TimBot to the other side of the grid.

The simulation takes place over a number of rounds. Each round consists, of seven phases:

1. **Start** each round by using one Jolt of energy to power the TimBot through the rest of the round. (Additional energy may be required to perform various actions. If a TimBot does not have a Jolt of energy, it becomes nonfunctional.
2. **Sense** the districts to the north, south, east, and west of their present district. The sensors can sense (i) the ripeness of Spresso plants in a district, and (ii) if there is a TimBot in a district.

3. **Fire** its ion cannon zero or more times (as long as the TimBot has the energy to do so). One ion shot requires one Jolt of energy. The ion cannon has a range of one district, to the north, south, east, or west of the TimBot.
4. **Defense** from ion cannon fire. If a TimBot detects an ion attack, it can defend itself with its force shield (as long as it has the energy to do so). It requires one Jolt of energy to stop one ion shot. If a TimBot does not have the energy to stop an ion shot, it is rendered nonfunctional.
5. **Move** to another district. A TimBot may move to the district north, south, east, or west of their present location. A move takes one Jolt of energy. A TimBot may also choose not to move.
6. **Battle** within a district. If two or more TimBots are in the same district, they use their shields to try to push the other TimBots out. The TimBots keep raising their shields until they run out of energy and become nonfunctional, or the other TimBot(s) run out of energy first. Hence, the TimBot with the greatest amount of energy survives and all the other TimBots become nonfunctional. The winner's energy level is decreased by the energy level of the runner up. Hence, if there is a tie for highest energy level, all combatants become nonfunctional.
7. **Harvest** the district (if there are Spresso plants to harvest). Once the Spresso plants are harvested, it takes a number of rounds to grow the Spresso plants.

If a TimBot's level drops below 0, they become nonfunctional.

Types of TimBots

There are four different personalities of TimBots:

ChickenBot : (Personality C) If a ChickenBot senses a TimBot in an adjoining district, it will move to an empty district that (preferably) has Spresso to harvest. That is, during the Move Phase, if the Chicken Bot has one or more Jolts of energy, it will consider all five districts (CURRENT, NORTH, EAST, SOUTH, and WEST), and will order them according to the following criteria (in order of preference):

1. There are no other TimBots in the district.
2. There are no TimBots in the adjacent districts.
3. The least number of rounds before Spresso plants are ready for harvest.
4. The order the districts as listed above.

It will then move to the most preferred district. The ChickenBot will never fire its ion cannon, preferring to conserve energy.

SpressoBot : The SpressoBot will always follow the Spresso plants. It will try to move to a district where it can harvest the Spresso plants as quickly as possible. That is, during the Move Phase, if the SpressoBot has one or more Jolts of energy, it will consider all five districts (CURRENT, NORTH, EAST, SOUTH, and WEST), and will order them according to the following criteria (in order of preference):

1. The least number of rounds before Spresso plants are ready for harvest.
2. There are no other TimBots in the district.
3. There are no TimBots in the adjacent districts.
4. The order the districts as listed above.

It will then move to the most preferred district. The SpressoBot will never fire its ion cannon, preferring to conserve energy.

AngryBot : The AngryBot will attack other TimBots. It will try to move to a district where it can attack another TimBot and, where there are Spresso plants ready for harvest (preferably). That is, during the Move Phase, if the AngryBot has three or more Jolts of energy, it will consider all five districts (CURRENT, NORTH, EAST, SOUTH, and WEST), and will order them according to the following criteria (in order of preference):

1. There are other TimBots in the district.
2. The least number of rounds before Spresso plants are ready for harvest.
3. The order the districts are listed above.

It will then move to the most preferred district.
If the AngryBot had less than three Jolts of energy, it behaves like a SpressoBot. An AngryBot is a subclass of SpressoBot.)

BullyBot : A BullyBot is a gun-happy ChickenBot. (A BullyBot is a subclass of ChickenBot.) During the Fire Phase, the BullyBot will shoot its cannon once into every adjoining district that has a TimBot in it, in the order NORTH, EAST, SOUTH, WEST, as long as long as it has three or more Jolts of energy. I.e., if its energy level drops to 2, it stops shooting. Otherwise, it behaves like a ChickenBot.

The Simulation

The simulator reads in the simulation configuration and then instantiates and runs the the simulation, printing out the state of the simulation after each round. The simulation continues until the required number of rounds have transpired.

Input

The input format to the simulator is as follows: The input comprises six integers, follow by one or more TimBot configurations. The first six integers are:

R : number of rows in the grid representing planet DohNat
C : number of columns in the grid representing planet DohNat
J : number of jolts that Spresso harvest yields
G : number of rounds needed for Spresso to grow after harvest
N : maximum number of rounds to run the simulation for
T : number of TimBot configurations to follow.

A TimBot configuration consists of one string and four integers.

- **P** : string denoting personality of TimBot. E.g., `chicken`, `spresso`, `angry`, `bully`.
- **I** : the numeric ID of the TimBot. (All IDs are assumed to be unique.)
- **X** : the x coordinate of the TimBot in the grid, where $0 \leq X < C$
- **Y** : the y coordinate of the TimBot in the grid, where $0 \leq Y < R$
- **E** : number of Jolts the TimBot initially has at the start of the simulation.

All values are separated by white-space.

Semantics

Upon reading the configuration, the simulator instantiates a *DohNat* using arguments R, C, J, and G. The simulator creates a list of *TimBot* objects, one for each configuration, and adds the objects to the *DohNat* object. The simulator then loops for at most N times, executing a round during each iteration. At the end of each round, the simulator outputs the state of the similation and traverses the list of *TimBot* objects to count all functional TimBots. If the number of functional TimBots is one or less, the simulator stops the loop.

Output

After each round completes, the simulator prints out to the console the string `Round` r, where r is the current round, followed by the grid of *DohNat.* The format is:

`Round` r

$$\begin{array}{lllll}
D_{0,0} & D_{1,0} & D_{2,0} & \dots & D_{C-1,0} \\
D_{0,1} & D_{1,1} & D_{2,1} & \dots & D_{C-1,1} \\
D_{0,2} & D_{1,2} & D_{2,2} & \dots & D_{C-1,2} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
D_{0,R-1} & D_{1,R-1} & D_{2,R-1} & \dots & D_{C-1,R-1}
\end{array}$$

where $D_{x,y}$ is the district at coordinates (x, y), and uses the format specified by the *District* class.

Sample Input

```
5 5 5 10 100 4
chicken 1 2 2 1
spresso 2 3 3 1
angry 3 2 3 1
bully 4 3 2 1
```

Sample Output

```
Round 0
|           0||           0||           0||           0||           0|
|           0||           0||           0||           0||           0|
|           0||           0||(C  1  5) 10||(B  4  5) 10||           0|
|           0||           0||(A  3  5) 10||(S  2  5) 10||           0|
|           0||           0||           0||           0||           0|
Round 1
|           0||           0||           0||           0||           0|
|           0||           0||(C  1  7) 10||(B  4  6) 10||           0|
|           0||           0||(A  3  3)  9||           9||           0|
|           0||           0||           9||           9||(S  2  7) 10|
|           0||           0||           0||           0||           0|
Round 2
|           0||           0||(C  1  9) 10||(B  4  8) 10||           0|
|           0||           0||           9||           9||           0|
|           0||(A  3  6) 10||           8||           8||(S  2 10) 10|
|           0||           0||           8||           8||           9|
|           0||           0||           0||           0||           0|
Round 3
|           0||           0||           9||           9||           0|
|           0||(A  3  9) 10||           8||           8||(S  2 13) 10|
|           0||           9||           7||           7||           9|
|           0||           0||           7||           7||           8|
|           0||           0||(C  1 11) 10||(B  4 10) 10||           0|
Round 4
|           0||(A  3 12) 10||           8||           8||(S  2 16) 10|
|           0||           9||           7||           7||           9|
|           0||           8||           6||           6||           8|
|           0||           0||           6||           6||           7|
|           0||(C  1 13) 10||           9||           9||(B  4 12) 10|
Round 5
|(S  2 18) 10||           9||           7||           7||           9|
|           0||           8||           6||           6||           8|
|           0||           7||           5||           5||           7|
|           0||(C  1 16) 10||           5||           5||           6|
|(B  4 14) 10||(A  3 10)  9||           8||           8||           9|
Round 6
|           9||           8||           6||           6||           8|
|(S  2 20) 10||           7||           5||           5||           7|
|           0||           6||           4||           4||           6|
|(C  1  8) 10||(A  3  7)  9||           4||           4||           5|
|           9||           8||           7||           7||           8|
Round 7
|           8||           7||           5||           5||           7|
|           9||           6||           4||           4||           6|
|(S  2 16) 10||           5||           3||           3||           5|
|(A  3  5)  9||           8||           3||           3||           4|
|           8||           7||           6||           6||           7|
Round 8
|           7||           6||           4||           4||           6|
|           8||           5||           3||           3||           5|
|(A  3  3)  9||(S  2 14)  4||           2||           2||           4|
|           8||           7||           2||           2||           3|
|           7||           6||           5||           5||           6|
Round 9
|           6||           5||           3||           3||           5|
|           7||           4||           2||           2||           4|
|           8||           3||(S  2 12)  1||           1||(A  3  1)  3|
|           7||           6||           1||           1||           2|
|           6||           5||           4||           4||           5|
Round 10
|           5||           4||           2||           2||           4|
|           6||           3||           1||           1||           3|
|           7||           2||(S  2 11)  0||           0||(A  3  0)  2|
|           6||           5||           0||           0||           1|
|           5||           4||           3||           3||           4|
Round 11
|           4||           3||           1||           1||           3|
|           5||           2||           0||           0||           2|
|           6||           1||(S  2 15) 10||           0||           1|
|           5||           4||           0||           0||           0|
|           4||           3||           2||           2||           3|
```