



Level 4



Glados:

Eh, a little impressed, I have to admit, but let's check out how *pros* build tower defense systems, maybe you can learn a thing or two from them.

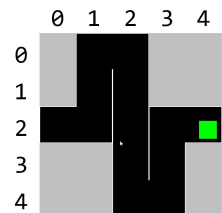
Now that we know exactly how they plan to attack, let's see how well a predefined defense system would do against them. We have great algorithms who planned our defense. It should be more than fine, we're confident they did good.

Task for Level 4:

Given a predefined set of basic towers, simulate how well they performed

We are interested a few aspects of the game. First of all, let's talk about **ending conditions**:

- when the first alien reaches the base, all hope is considered to be lost and we have failed
 - the base is defined as the next cell after the last one from the list at level 2
 - the red square is the base
 - when any alien tries to move to the red square, the game is over
- when all aliens are dead, all hope is regained and we have won



We will want to know in which **tick** will the **simulation end** and what the result will be. The ticks are **0 indexed**



Let's talk about our defense mechanism, the **Towers**.

- › While we have blueprints for multiple types of towers, due to limited resources we can only use one type
- › The type of tower is characterized by:
 - › **DAMAGE**: float value equal to the amount of hit points removed from an alien on hit
 - › **RANGE**: the radius of the circle, centered on the tower, in which it fires and looks for targets
- › Additionally, the tower will have a position, (x, y) coordinates of where it will be positioned on the grid
- › For this simulations, you have **T** towers and they **spawn at tick 0**



We need to also update our definition of an **Alien**

- › While we have classified multiple types of aliens, we will only simulate with one
- › Each type is characterized by:
 - › **HEALTH**: the amount of hit points they spawn with
 - › **SPEED**: the amount of cells they move per tick
- › If at the end of a tick, **HP** of an **alien** is **≤ 0** , it becomes **dead**



- › Each tick of the simulation should
 1. update the alien positions
 2. check if any alien has reached the end*
 3. spawn new aliens
 4. simulate tower shots
 5. check for dead aliens
 6. check if all aliens are dead and no more will be spawning*

- › **IN THIS EXACT ORDER!**
- › * - the tick ends if the check is true and the game is over
- › **Tower shots happen before tower spawn => In tick 0, no tower shots are fired**



- › A tower has two states
 - › **LOCKED**: if it has a target
 - › **SEEKING**: otherwise
- › A tower shot consists of
 - › check if it is locked and the locked in target is valid
 - › if not, it looks for the closest alien that is in range. In case of ties, chooses the one with the lowest id
 - › if it had a valid target or has just found a new one, deal damage to target
 - › subtracts hit points from the target health according to the damage value
- › An alien is in range of a tower at tick t if the **euclidean distance** between the alien's position at tick t and the tower's position is **less than or equal** to the tower's range
- › A valid target is an alien that is both in range of the tower and alive



Level 4

- › While this may not seem like the optimal strategy to use, this is the one used by our simulations. In order to pass this level you will need to implement it according to our specs
- › Again, towers don't switch to the closest target, if they are locked to a different one!
- › **All towers keep their currently locked target until target dies or is out of range**
- › **All towers fire at once, which means that if multiple towers have the same target, they might overkill it!**

Input format:

<WX> <WY>

<X> <Y>

<Movement_Sequence>

<Health> <Speed>

<N>

<SpawnTime>

...

<SpawnTime>

N Times

<Damage> <Range>

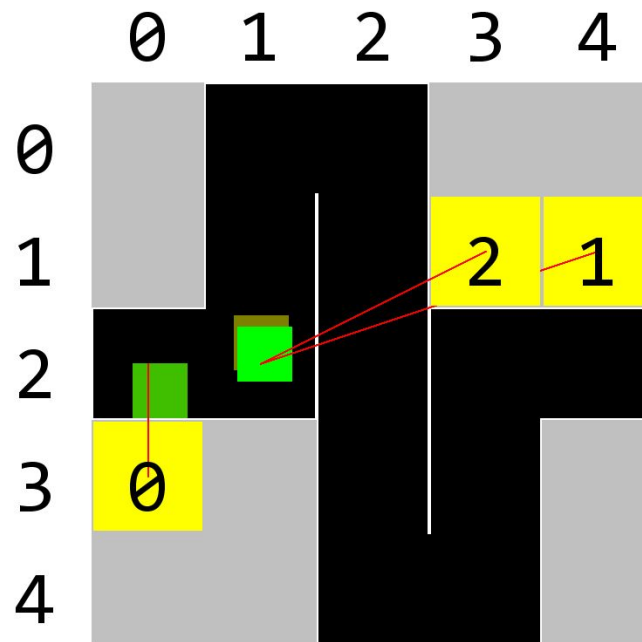
<T>

<X> <Y>

...

<X> <Y>

T times



*Click to see animated version

Input example:

5 5

0 2

F 1 T 3 F 2 T 1 F 1 T 1 F 4 T 3 F 1 T 3 F 2 T 1 F 1

40.0 1.0 -> aliens have 40 health and speed 1

10 -> there will be 10 aliens

5

4

3

4

2

1

5

6

5

2

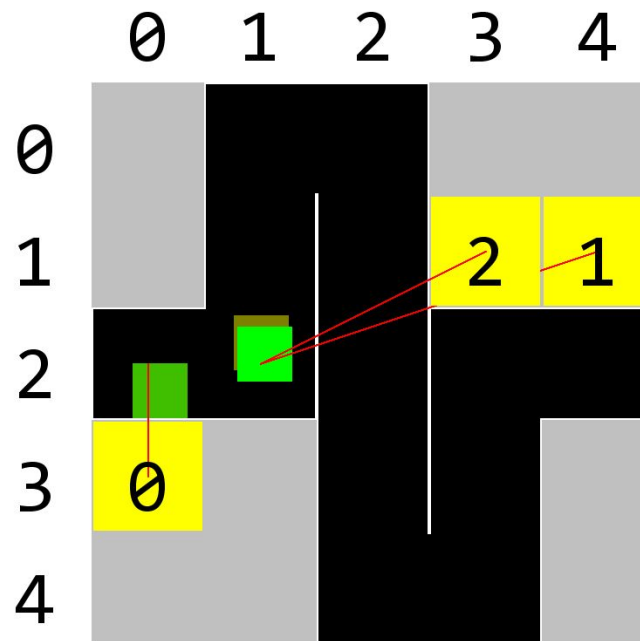
10.0 5 -> towers deal 10 damage and have a range of 5

3 -> there will be 3 towers

0 3 -> at x = 0 y = 3

4 1 -> at x = 4 y = 1

3 1 -> at x = 3 y = 1



*Click to see animated version

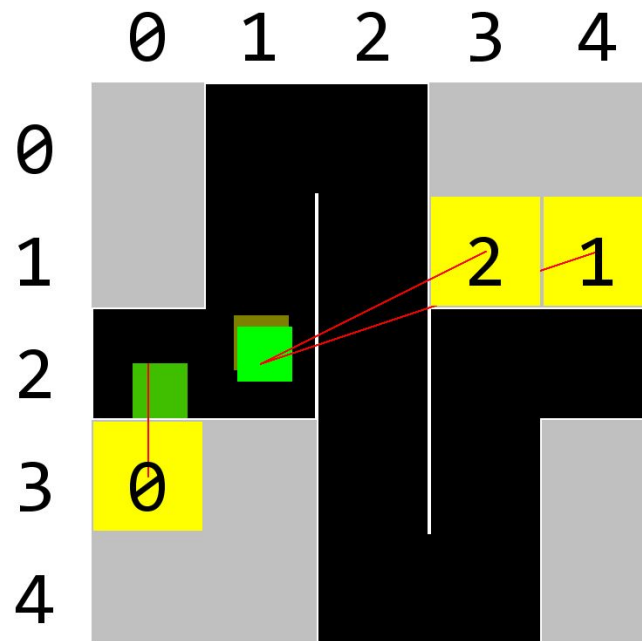
Output format:

<Simulation_Time>

<Simulation_Result>

Simulation_Time = Number of ticks until simulation ends

Simulation_Result = "WIN" if all aliens die or "LOSS" if any alien reaches our base



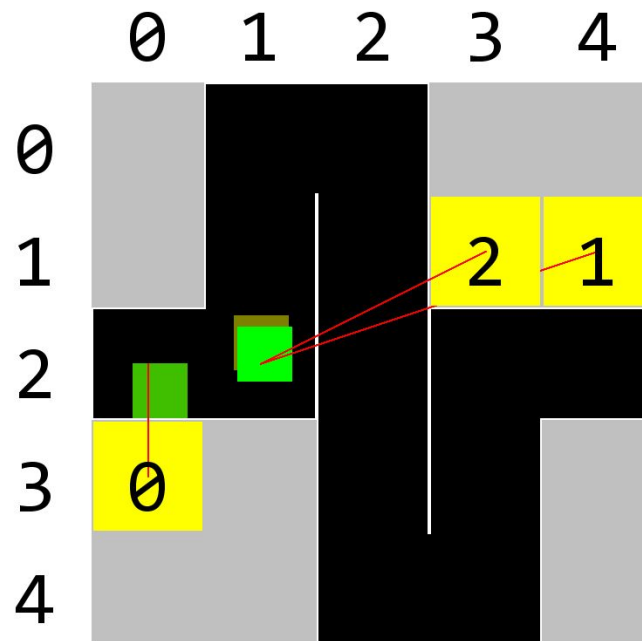
*Click to see animated version

Output example:

15

WIN

At the end of tick 15, all aliens were killed by the towers



*Click to see animated version