**This is a league based challenge.**

For this challenge, multiple leagues for the same game are available. Once you have proven yourself against the first Boss, you will access a higher league and harder opponents will be available.

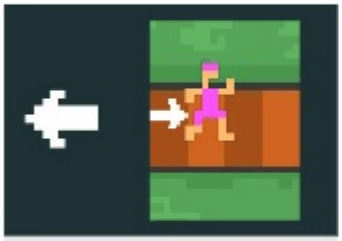
**NEW:** In wood leagues, your submission will only fight the boss in the arena. Win a best-of-five to advance.

## Goal

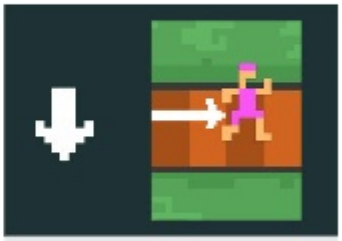
End the game with a higher **score** than your opponent.

## Rules

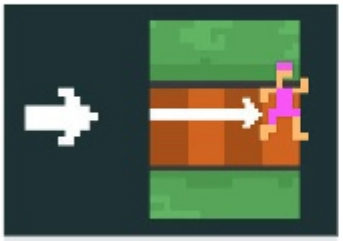
Play multiple runs of the **Hurdle Race mini-game** against two other players.
To play, print either `LEFT`, `DOWN`, `RIGHT`, `UP` on each turn.



`LEFT`: move forward by `1` space.



`DOWN`: move forward by `2` spaces.



`RIGHT`: move forward by `3` spaces.

`UP`: jump over one space, ignoring any hurdle on the next space and moving by `2` spaces total.

Jump over hurdles or you will **collide** with them, causing your agent to be **stunned** for `3` turns.

The race track is `30` spaces long, players begin on space `0`. When a player reaches the end, the race ends. Two things will then occur:

- According to their position on the race track, each player is awarded a **gold**, **silver** or **bronze** medal.
- The mini-game **resets**, this means that for one turn all input is ignored.

After `100` turns, your **final score** is `nb_silver_medals + nb_gold_medals * 3`.

The mini-game is running on an **old arcade machine**. Your program will receive the `8` **registers** used internally by the machine: `GPU`, containing a string and `reg0` to `reg6` containing integers. What those values represent specific to the game being played.

In this case:

| Register | Description | Example |
|---|---|---|
| GPU | ASCII representation of the racetrack. `.` for empty space. `#` for hurdle. | `.....#...#...#.....` |
| reg0 | position of player 1 | 0 |
| reg1 | position of player 2 | 6 |
| reg2 | position of player 3 | 12 |
| reg3 | stun timer for player 1 | 1 |
| reg4 | stun timer for player 2 | 0 |
| reg5 | stun timer for player 3 | 2 |

| | |
|---|---|
| reg6 | *unused* |

The **stun timer** is the number of turns remaining of being stunned (`3`, then `2`, then `1`). `0` means the agent is not stunned.

During a **reset** turn, the `GPU` register will show `"GAME_OVER"`.

**Victory Condition**

You have a higher **final score** after `100` turns.

**Defeat Condition**

Your program does not provide a command in the allotted time or it provides an unrecognized command.

## Debugging tips

- Press the gear icon on the viewer to access extra display options.
- Use the keyboard to control the action: space to play/pause, arrows to step 1 frame at a time.

## Game Protocol

### Initialization Input

**First line:** `playerIdx` an integer to indicate which agent you control in the mini-games.
**Next line:** the number of simultaneously running mini-games. For this league it's `1`.

### Input for One Game Turn

**Next `3` lines:** one line per player, ordered by `playerIdx`. A string `scoreInfo` containing a breakdown of each player's final score. In this league, it contains `4` integers. The first integer representing the player's current **final score points** followed by: `nb_gold_medals`, `nb_silver_medals`, `nb_bronze_medals`.

**Next `nbGames` lines:** one line for each mini-game, containing the eight space-separated registers:

- `gpu` a string
- `reg0` an integer
- `reg1` an integer
- `reg2` an integer
- `reg3` an integer
- `reg4` an integer
- `reg5` an integer
- `reg6` an integer

Unused registers will always be `-1`.

### Output

One of the following strings:

- `UP`
- `RIGHT`
- `DOWN`
- `LEFT`

### Constraints

$0 \leq$ `playerIdx` $\leq 2$
$1 \leq$ `nbGames` $\leq 4$ *(across all leagues)*

Response time per turn $\leq 50$ ms
Response time for the first turn $\leq 1000$ ms

### What is in store for me in the higher leagues?

- 4 hurdle race mini-games will be played simultaneously
- 4 entirely different mini-games will be played simultaneously!