

Robots

A large group of angry robots are chasing you, as the sole survivor of the robocolypse, must survive to warn the others. You have a scanner which can detect robots from 5 squares away from you (whatever a square means), however don't let them get too close or human kind as we know will cease to exist! Try to trick them and get to the exit.

Enough with the glamorous introduction, you are to write a program that will have a random number of robots placed (however they must be placed at least 2 spaces away from the human).

Map

The map will hold the location of the robots, human and exit. Each map must mandate an exit the human must reach to proceed into the next level. Once all robots are destroyed or the human has reached the exit, the player will be moved to the next level.

Every time the human escapes, a new map will need to be generated. Specifying the starting location of the human, how many robots and each robot's starting location.

Each 8x8 map, will denote a robot using an R symbol and the human will be denoted using a H symbol. When two robots have collided the cell remaining is D, which shows that the robots have been destroyed on that location. Exits are shown using the E symbol.

Example of a map.

1								
2							E	
3			H					
4								
5							R	
6					D			
7								
8			R			R		
9								

Robots

Each robot will have a position on the board which is specified from user input. A robot can be destroyed when two robots collide by occupying the same cell on a move. A robot will move towards the human until it has captured it. Once it has captured the human, the robots have won. A robot can move in 4 directions, to determine where they move, requires determining the angle between the robot and the human or finding axis with the greater distance between the . If distance between the robot and the human is greater on the x axis, the robot will move towards human on the x axis, likewise on the y axis.

Example:

1		F		A	
2					
3	E			I	C
4			-----R-----		
5		H			D
6					
7			G		B

Where the robot will move based on each point

1	A : robot.y - 1;
2	B : robot.y + 1;
3	C : robot.x + 1;
4	D : robot.x + 1;
5	E : robot.x - 1;
6	F : robot.y - 1;
7	G : robot.y + 1;
8	H : robot.x - 1;
9	I : ?

What about 45 degrees or when the robot and human are diagonally across from each other (I). It will use the information from the human's previous location to determine where it should move. If the human moved on the y axis, the robot will move on the x.

Example:

x, y, w and v are instances of the previous location where the robot was.

1	y	r: 2 2
2	x H w	h: 1 1
3	v R	

Decision:

```
1 | x : robot.y - 1;
2 | w : robot.y - 1;
3 | y : robot.x - 1;
4 | v : robot.x - 1;
```

Human

The human will move around the map and the program will take commands from the user. The following commands will be accepted by the program to move the human:

- North (NT)
- South (ST)
- West (WT)
- East (ET)

Exit Location

The exit location will use a number provided by via command line arguments. This will act as a seed for the random number generated. Use `srand()` to set the seed.

After setting the seed, you can set the exit coordinates using the `rand()` function. Use the following statement for computing the point for the exit.

```
1 | x = rand() % 8;
2 | y = rand() % 8;
```

Commands

Your program must support the following commands

- move <(NT|ST|WT|ET)> - Moves the human to a new cell.
- display - Displays the grid and where the robots, human and exit is.
- exit - Quits the program

Examples

Example 1 (Map setup)

```

1 Human starting point: 0 3
2 Number of robots: 3
3 Robot 1: 4 4
4 Robot 2: 7 2
5 Robot 3: 0 0

```

Example 2 (Display)

```

1 display
2
3 | | | | | | | |
4 | | | | | | E |
5 | | H | | | | |
6 | | | | | | | |
7 | | | | | | R |
8 | | | | D | | |
9 | | | | | | | |
10 | | R | | | R | | |
11

```

Example 3 (Move and Display)

```

1 display
2
3 | | | | | | | |
4 | | | | | | E |
5 | | H | | | | |
6 | | | | | | | |
7 | | | | | | R |
8 | | | | D | | |
9 | | | | | | | |
10 | | R | | | R | | |
11
12 move ET
13
14 display
15
16 | | | | | | | |
17 | | | | | | E |
18 | | H | | | | |
19 | | | | | | | |
20 | | | | | | R |
21 | | | | D | | |
22 | | R | | | R | | |
23 | | | | | | | |
24

```

H: 1 2
 R1: 6 4
 R2: 4 7
 R3: 1 7

seed 3
 check : when R move at E

Example 4 (Move And Exit)

```
1 display
2
3 | | | | | | | |
4 | | |E|H| | | |
5 | | | | | | | |
6 | | | | | | | |
7 | | | | | |R| |
8 | | | | |D| | |
9 | | | | | | | |
10 | |R| | |R| | |
11
12 move WT
13
14 Well done! You survived, but it isn't over!
15
16
17 Human starting point: 0 0
18 Number of robots: 3
19 Robot 1: 4 4
20 Robot 2: 7 2
21 Robot 3: 5 5
22
23 ...
24
```

Example 5 (Move And Robot)

```
1 display
2
3 | | | | |H| | |E|
4 | | | | | |R| | |
5 | | | | | | | |
6 | | | | | | | |
7 | | | | | | | |
8 | | | | |D| | |
9 | | | | | | | |
10 | |R| | |R| | |
11
12 move ET
13
14 The robots got you! We are doomed!
15
16 Human starting point: 0 0
17 Number of robots: 3
18 Robot 1: 4 4
19 Robot 2: 7 2
20 Robot 3: 5 5
```

H: 4 0
R: 5 1

21	
22	...
23	

Error Examples

Example 1 (Invalid Coordinates)

1	Human starting point: 9 H
2	Invalid Coordinates, Try again
3	Human starting point: 3 0
4	Number of robots: 3
5	Robot 1: 4 4
6	Robot 2: T 2
7	Invalid Coordinates, Try again
8	Robot 2: 7 2
9	Robot 3: 0 0

Example 2 (Invalid No Robots)

1	Human starting point: 3 0
2	Number of robots: G
3	Invalid number, Try again
4	Number of robots: 3
5	Robot 1: 4 4
6	Robot 2: 7 2
7	Robot 3: 0 0

Example 3 (Invalid Move)

1	move FT
2	Invalid