**17. Nick Needs A Ride**

# Program Name: Nick.java Input File: nick.dat

Nick wants to attend computer science club after school, but he doesn’t own a car and his parents have more important things to do than pick him up. He systematically begs every person in computer science for a ride, but everyone is “busy” and unable to take him home. He is allergic to late buses so his only option is to suck it up and walk home. Unfortunately, Nick gets hungry very quickly during physical exertion. He starts his journey with a certain amount of hunger points. Every step he takes reduces his hunger points by a certain amount. Finding a trash can will replenish a specified amount of hunger points, because Nick can nourish himself off the scraps of food inside. If his hunger points ever reach zero, he collapses and doesn’t make it home (unless Nick arrives at home or a fresh trash can at 0 hunger points, then he will survive).

Write a program to figure out if Nick arrives home safely or has to spend the night on the side of the road.

**Input**

The first number, n, will be the number of test cases.

Each test case has 4 integers on the first line, followed by r number of lines.

The first integer of each test case, r, will be the number of rows of the map.

The second integer of each test case, d, will be the amount of hunger subtracted from each step Nick takes.

The third integer of each test case, t, will be the amount of hunger gained upon reaching a new trash can.

The fourth integer of each test case, s, will be the amount of hunger points Nick starts with.

‘S’ marks the school, or Nick’s starting point.

‘H’ marks his house, or Nick’s destination. If Nick arrives at home at 0 hunger points, he will survive.

‘T’s are trash cans, where Nick can replenish his hunger points by t. If Nick arrives at a trash can at 0 hunger points, he will survive. Nick can traverse through trash can tiles. Once a trash can is visited, it will not contain any food left (Nick cannot return to the same trash can and regain health points again).

‘@’s are obstacles that Nick cannot traverse through. The map will always be surrounded by walls.

‘.’s are empty spaces that Nick can travel through.

Nick can only travel north, east, south, or west.

**Output**

If Nick makes it to his house, print out STILL ALIVE. If Nick does not make it to his house, print out RIP IN PEPRI

**Example Input File**

3

4 2 10 30

@@@@@@@

@....S@

@H....@

@@@@@@@

10 5 50 60

@@@@@@@@@@

@@@@@@.....T...@

@...@..........@

@.S.@...........@@@@@@@@@@@

@.................T.@H@..T@

@.....................@...@

@........T...............@

@............@@@@@@@@@@@@

@...........@

@@@@@@@@@@@@

(continued on next page...)

4 2 10 30

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

@................T....T......T......T....T....S@

@H....T........................................@

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

**Example Output to Screen**

STILL ALIVE

STILL ALIVE

RIP IN PEPRI