

## Challenge 6 - Ice Cave

You are trapped inside a dark cave. You are freezing, and you realize that you won't last much longer if you don't escape. The cave's floor is covered in ice, so if you try to move, you will slide until something (a rock, a wall...) stops you. You need to find a way out of the cave if you want to survive, and you want to figure out how long it will take you.

### Input

The first line of the input is an integer,  $N$ .  $N$  test cases follow. For each test case, you will be given the width and length  $W$  and  $H$  (in meters) of the cave, the speed  $S$  (in  $m/s$ ) that you will have while sliding, and the time  $T$  (in seconds) that you will need to make yourself move after you stop. Those values are contained in a single line, separated by whitespaces. You will also have a rudimentary map of that cave.

### Output

For each test case, output the time in seconds (rounded, half a second won't make any difference in your survival options) that you will need to escape the cave.

**Note:** You can only move up, down, right or left, not diagonally. And while you are sliding, you have absolutely no control so you can't turn or stop.

### Map legend

```
• Ice
# Obstacle (wall, rock, whatever)
X Your starting position
O (Uppercase "o") Exit
```

The exit is always on the border of the cave, and will always be reachable.

### Example test case

```
4 5 1 3
####
#X•#
#••#
#••O
####
```

Here, the map dimensions are **4x5**, you slide at **1m/s**, and you have to spend **3** seconds to start sliding after you stop (or before you start).

### Sample input

```
2
4 5 1 3
```

### Sample output

10  
19

PS: Yes, we've played Pokemon Gold/Silver. Deal with it.

## Submit & test your code

To test and submit code we provide a set of tools to help you. Download [contest tools](#) if you haven't already done that. You will then be able to test and submit your solution to this challenge with the challenge token.

Challenge token: pUGr911Tbs06LSAfQrws

## To test your program

```
./test challenge pUGr9l1Tbs06LSAfQrwS path/program
```

A nice output will tell you if your program got the right solution or not. You can try as many times as you need.

## To submit your program to the challenge

```
./submit challenge pUGr9l1Tbs06LSAfQrwS path/source pkg.tgz path/program
```

Note that you first need to solve the test phase before submitting the code. During the submit phase, in some problems, we might give your program harder questions, so try to make your program failsafe.

**Important:** In this phase, you must provide the source code used to solve the challenge and, if necessary, a brief explanation of how you solved it.

Remember **you can only submit once!** Once your solution is submitted you won't be able to amend it to fix issues or make it faster, so please be sure your solution is finished before submitting it.

If you have any doubts, please check the [info section](#).

## Go ahead

### I'm done! :)

Once you have submitted your code, hit refresh and continue to next challenge.

### I'm stuck! :(

Be sure you follow the [Tuenti Engineering](#) twitter for updates and possible hints during the contest.

If this challenge is too hard and you are blocked, you will be able to skip it after two hours. Note that **you won't be able to complete it later**, and you have a limited number of challenges to skip.

Finally, if you run out of skips but are still really stuck with one problem, you will be able to skip it after 24 hours.

### Challenge status:

Test case	Not done
Solution submitted	Not done
Skip	You still have to wait 0h, 30m and 0s to be able to skip this challenge

Refresh status

Tweet about this! [#TuentiChallenge3](#)

 Share

Follow [@Tuentieng](#)