

ECOO '16 R2 P4 - Hop, Skip and Jump

Time limit: 20.0s **Memory limit:** 256M

Hop, Skip and Jump are brothers seeking valuable items (the Cup, the Flag and the Treasure) in a 2D world of ladders and platforms. Their world is a grid of characters and each location on the grid is one of three possible types: a platform (the `=` character), a ladder (the `#` character) or clear (the `.` character). A brother can never occupy a location that has a platform in it, but he can occupy any location that is clear or contains a ladder. A brother can also stand in the location immediately above a platform or a ladder, as long as the location they are occupying is clear or contains a ladder.

Basic Movement Rules

All three brothers are affected by gravity. A brother is falling if he is currently on a clear location and the location beneath him is also clear. A falling brother always moves down until the location below him contains a platform or a ladder. A brother can fall an infinite distance without getting hurt. All three brothers can move one spot left or right, as long as they are not falling and as long as the location they are moving to does not contain a platform. All three brothers can climb up as long as their current location contains a ladder and the location above them does not contain a platform. All three brothers can climb down if the space they are occupying contains a ladder and the space below them does not contain a platform. All three brothers can also climb down if the space below them contains a ladder and they are not falling.

Here are some examples (`h` = Hop, `s` = Skip, `j` = Jump):

a	b	c	d	e	f	g
<code>#h.</code> <code>===</code>	<code>=s.</code> <code>=#.</code>	<code>.j.</code> <code>=#.</code>	<code>.#.</code> <code>.h.</code> <code>.#.</code>	<code>=..</code> <code>=s=</code> <code>=#=</code>	<code>.#.</code> <code>.j.</code> <code>...</code>	<code>===</code> <code>.h.</code> <code>.#.</code>

In example **a**, Hop can move left or right.

In example **b**, Skip can move right but not left.

In example **c**, Jump can't move left or right because he is falling.

In example **d**, if Hop's location contains a ladder, he can climb up or down, otherwise he can only climb down.

In example **e**, if Skip's location contains a ladder, he can climb up or down, otherwise he can only climb down.

In example **f**, if Jump's location contains a ladder, he can climb up or down, otherwise he is falling.

In example **g**, Hop can only climb down regardless of whether or not his location contains a ladder.

Special Moves for Hop

As long as he isn't falling, Hop can hop up to the left or right. A hop moves Hop to a target location one spot sideways (left or right) and one spot up, provided the location immediately above Hop and the target location are both not a platform. Hop can also hop straight up (as long as he's not falling) provided the location above him isn't occupied by a platform.

Here are some examples:

a	b	c	d
#.. .h= ==.	=#. .h= ===	.=. =h= ===	.#. .h. .=.

In example **a**, Hop can hop up, left and right. In example **b**, Hop can hop up and right but not left.

In example **c**, Hop cannot hop in any direction. In example **d**, Hop can hop up, left and right.

Special Moves for Skip

As long as he isn't falling, Skip can skip to the left or right. A skip moves Skip to a target location two spots sideways (left or right), provided the target location and the intervening location are both not occupied by a platform.

Here are some examples:

a	b	c	d
==... ..S.. =.=.=	..=# .=S## ..=..S.= ..#.=S.. =...#

In example **a**, Skip can skip left or right. In example **b**, Skip can skip right but not left. In example **c**, Skip can skip left but not right.

In example **d**, if Skip's location contains a ladder, he can skip left and right, otherwise he is falling.

Special Moves for Jump

Jump can Hop and Skip just like his brothers. As long as he isn't falling, Jump can also jump to the left or right using a long jump or a high jump and he can jump up with a spring jump.

A long jump moves Jump to a target location three spots sideways (left or right), provided the target location and the intervening locations are all free of platforms.

A high jump moves Jump to a target location that is two spots sideways (left or right) and one spot up, provided the location beside Jump (in the direction of his jump), the location above Jump, the location above the location beside Jump, and the target location are all free of platforms.

A spring jump moves Jump two spots up, provided the target location and the intervening location are free of platforms.

Here are some examples:

a	b	c	d
<pre> ===#.#. ...J... =..=#.# </pre>	<pre> ...#...=. .=.J.=. ...=... </pre>	<pre> ...#...=. ..=J... </pre>	<pre> ...=...J... ...#... </pre>

In example **a**, Jump can long jump left and right, can high jump right but not left, and can spring jump.

In example **b**, Jump can't long jump at all and can't high jump right, but can high jump left or spring jump.

In example **c**, if Jump's location contains a ladder, he can long jump right and can spring jump but he can't long jump left or high jump in either direction. However, if Jump's location is clear, he is falling.

In example **d**, Jump can long jump and high jump in both directions, but cannot spring jump.

Input Specification

The input will contain 10 test cases.

The first line of each test case consists of two integers W and H ($0 < W, H < 200$) representing the width and height of the level.

The next H lines of data will consist of W characters, representing the layout of the level.

The three brother locations will be represented by lowercase **h**, **s** and **j**. Initially, the brothers always occupy a location that is clear. The three goal locations will be represented by an uppercase **C**, **F** and **T** (for Cup, Flag and Treasure). The remaining locations will each be one of the following: a **.** character (ASCII 46) for clear, a **#** character (ASCII 35) for a ladder, or an **=** character (ASCII 61) for a platform.

Output Specification

For each test case you must output three lines using only capital letters. Each line should consist of the name of a brother, followed by a space, followed by the first letter of each goal that brother is able to reach from their start position using their movement rules. The output for each test case should show `HOP` first, then `SKIP`, then `JUMP`. The goals each brother can reach should be listed in alphabetical order (`C`, then `F`, then `T`). Note that each goal that a brother can reach could be reachable via a completely different path from the others. If a brother cannot reach any of the goals print only that brother's name. You will be allowed 45 seconds of total execution time for all 10 test cases.

Note that the sample input below contains only 1 test case but the data files will contain 10. For readability, you can display your output in groups of 3 lines separated by a blank line.

Sample Input

```
10 10
#.....#.
#=#...#...#.
#..==..C.#.
..#..===#.
#.#.....##
..#...=#.#
.j..#...##
.=...s...#
=..h==T..F
...=.=...=
```

Sample Output

```
HOP T
SKIP T
JUMP CFT
```

Educational Computing Organization of Ontario - statements, test data and other materials can be found at ecocs.org