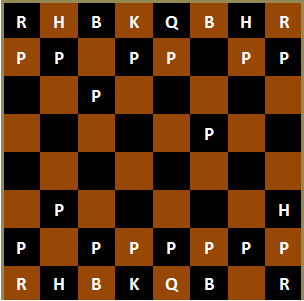
# Problem 4 – PHP Chess

The International Chess Federation is developing chess software for displaying real-time data about a chess game. They’ve asked SoftUni to recommend them someone gifted to join their project, and it appears you’re the lucky guy! Or, well, girl..

Your job is to write a PHP program that reads a **string representation of the chess board**. It then validates the chess board, converts it to an **HTML table** and prints **each of the available pieces’ count**.

Chess is a game that is played on a 8x8 board. The game is played with 6 unique pieces: **R**ook, **H**orseman (Knight), **B**ishop, **K**ing, **Q**ueen, **P**awn.

You are given a string, where each Latin letter is the first letter from a chess piece name. " " (space) denotes an empty position. For example:

*"R-H-B-K-Q-B-H-R/P-P- -P-P- -P-P/ - -P- - - - - / - - - - -P- - / - - - - - - - / -P- - - - - -H/P- -P-P-P-P-P-P/R-H-B-K-Q-B- -R"*

The positions are separated by dashes("-") from one another. The board rows are separated by a slash("/"). A cell can ONLY containthe letters **R**, **H**, **B**, **K**, **Q**, **P** (denoting the respective piece’s name) and " "(denoting an empty cell). Each row must contain exactly 8 cells. There must be exactly 8 rows. If any of the conditions is not met, print "Invalid chess board" in **<h1></h1>** heading tags.

If the input string represents a valid chess board, print it as an **HTML table**. (without the styling). The whole board should be put in **<table></table>** tags. Each row should be put in **<tr></tr>** tags. Each cell should be put in **<td></td>** tags, containing only the first letter of the respective piece or just space(" ") if the position is empty. (See the example below.)

Finally, each of the available pieces and their count is printed in the format **"[Piece full name]":[Count]** as a JSON string. The entries should be **sorted by keys** (piece name). Example:

{"Bishop":4,"Horseman":4,"King":2,"Pawn":16,"Queen":2,"Rook":4}

### Input

The input will be read from an **HTTP GET request**. The **input string** will be received from a **text** **input field with name 'board'**.

### Output

If the input string represents an invalid chess board (according to the rules mentioned above), print "Invalid chess board" in **<h1></h1>** tags. Otherwise, the output should consist of the **board as an HTML table** and each of **the pieces and their count as a JSON string**.

### Constraints

* The cells will always be separated by dashes ("-").
* The rows will always be separated by slashes("/").

### Examples

|  |  |  |
| --- | --- | --- |
| **Input** | | **Output** |
| board | R-H-B-K-Q-B-H-R/P-P- -P-P- -P-P/ - -P- - - - - / - - - - -P- - / - - - - - - - / -P- - - - - -H/P- -P-P-P-P-P-P/R-H-B-K-Q-B- -R | <table><tr><td>R</td><td>H</td><td>B</td><td>K</td><td>Q</td><td>B</td><td>H</td><td>R</td></tr><tr><td>P</td><td>P</td><td> </td><td>P</td><td>P</td><td> </td><td>P</td><td>P</td></tr><tr><td> </td><td> </td><td>P</td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td>P</td><td> </td><td> </td></tr><tr><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td><td> </td></tr><tr><td> </td><td>P</td><td> </td><td> </td><td> </td><td> </td><td> </td><td>H</td></tr><tr><td>P</td><td> </td><td>P</td><td>P</td><td>P</td><td>P</td><td>P</td><td>P</td></tr><tr><td>R</td><td>H</td><td>B</td><td>K</td><td>Q</td><td>B</td><td> </td><td>R</td></tr><table>  {"Bishop":4,"Horseman":4,"King":2,"Pawn":16,"Queen":2,"Rook":4} |
| **Input** | | **Output** |
| board | asdfghjkl | <h1>Invalid chess board</h1> |