Name _____ Period _____

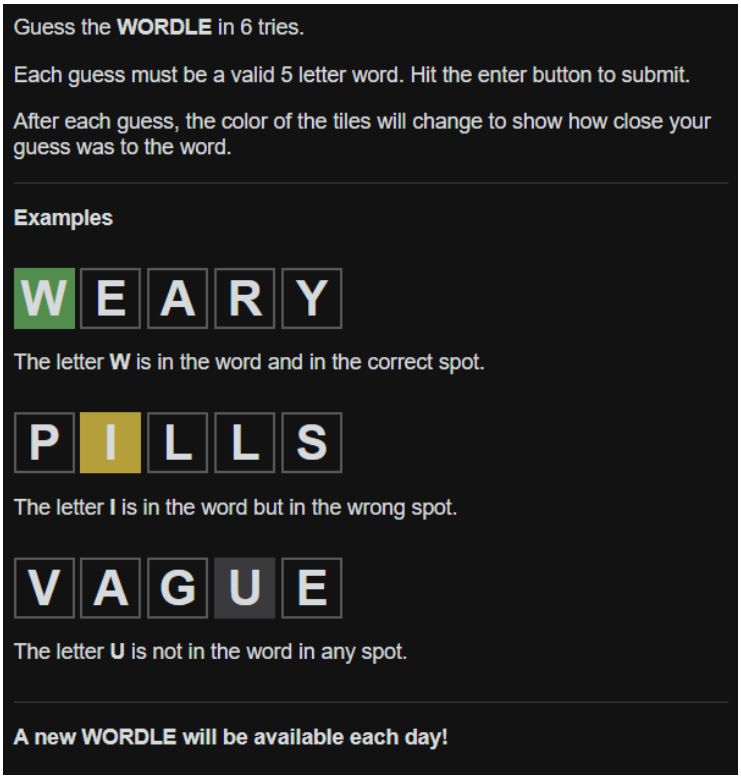
Wordle

Your Tasks (Mark these off as you go)

- ☐ Review how to play Wordle
- ☐ Write the *addLetter* function
- ☐ Write the *deleteLetter* function
- ☐ Write the *checkWord* function
- ☐ Receive credit for this lab guide

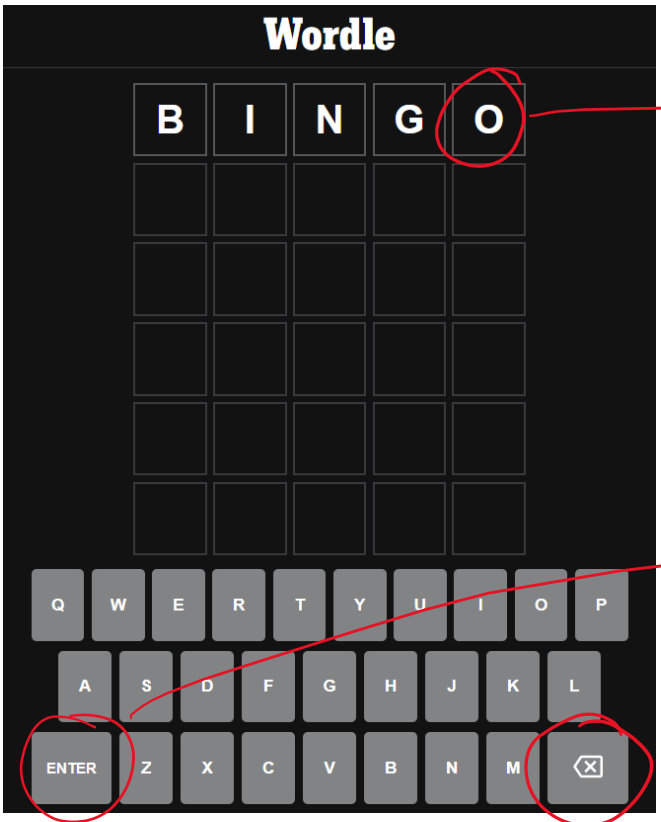
☐ Review how to play Wordle

The popular Wordle game is described below,



The interface for playing the game is shown below. The user guesses the word by clicking on the letters. Each letter that is clicked is displayed in a box. Once 5 letters are clicked, the user can click the Enter button and the guess is evaluated according to the rules described above. If the user makes a mistake, the user may also delete a letter by clicking the back button. The game also incorporates the following restrictions.

- The user may not enter more than 5 letters at a time
- The user may not delete letters beyond the current row



No more than 5 letters may be entered at a time.

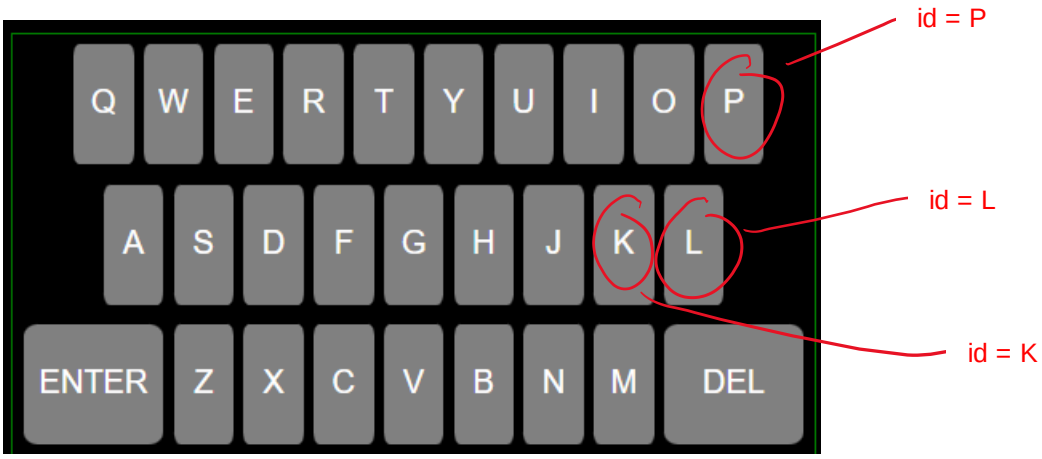
Once 5 letters are entered, the guess is evaluated by clicking the *Enter* button

The back button deletes a letter. Only letters on the current row may be deleted

☐ Write the *addLetters* function

Each letterBox that is displayed is assigned a numerical id as shown below. Notice, that the first id of each row is multiple of 5.

The id of each letterButton is the same as the letter it displays. Below are some examples.



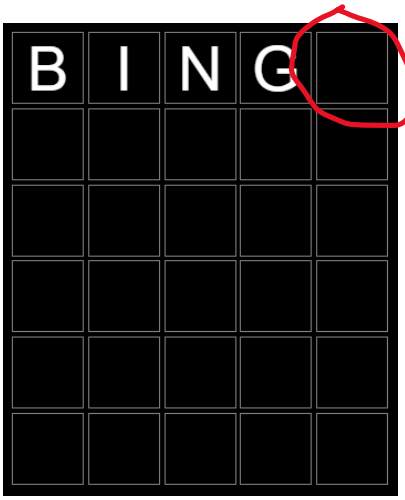
Each time a letterButton is clicked, the addLetter function is called and a letter is added to the next available letterBox. Letters may not be added beyond the current row. Complete the *addLetter* function below. The following can be used to capture the id of the letterButton clicked. Once the id is captured, it must be assigned to the innerHTML of the next available letterBox. Feel free to declare additional global variables as needed to keep track of the buttons clicked. A global *letterCount* variable might be useful for keeping track of each letter added. Each time a letter is added you can increment *letterCount*.

e.target.id

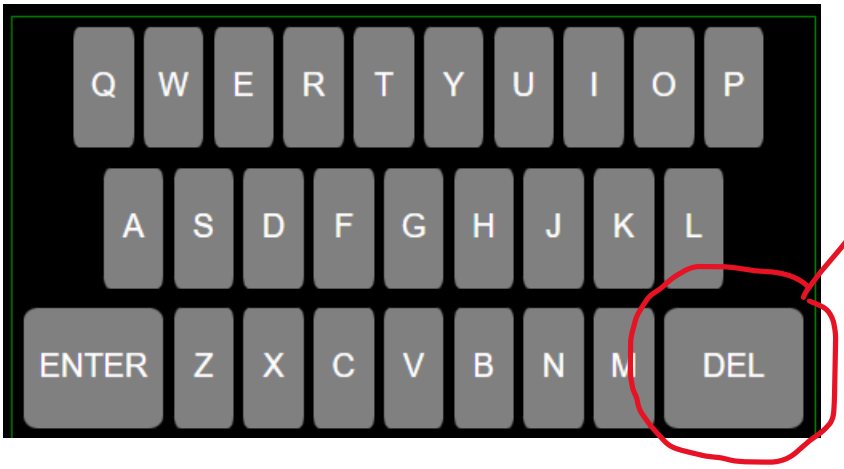
```
function addLetter(e){
    //complete the addLetter function below




}
```

□ Complete the *deleteLetter* function

When the delete button is clicked, the last letter that was selected must be removed.



When the delete button is clicked the last letter that was selected should be removed.



When the DEL button is clicked, the *removeLetter* function is called.

C is the last letter that can be removed. The O, because it is on the previous line, cannot be removed.

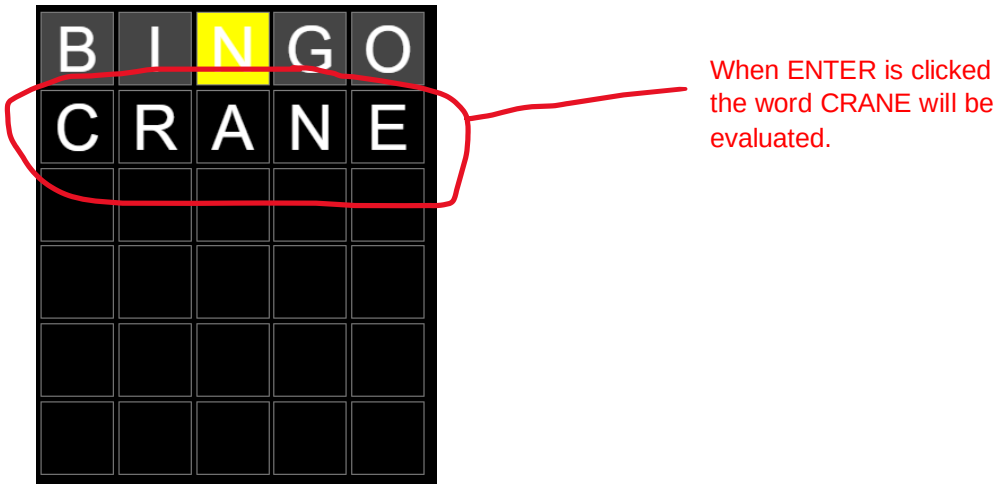
Each time the DEL button is clicked, the *removeLetter* function is called and the last letter that was entered is removed, that is the innerHTML of the corresponding letterBox is set to "". Letters that are not on the current row cannot be removed. Feel free to declare additional global variables as needed to keep track of the buttons clicked. A global *letterCount* variable might be useful for keeping track of each letter added and removed. Each time a letter is removed you can decrement the *letterCount*.

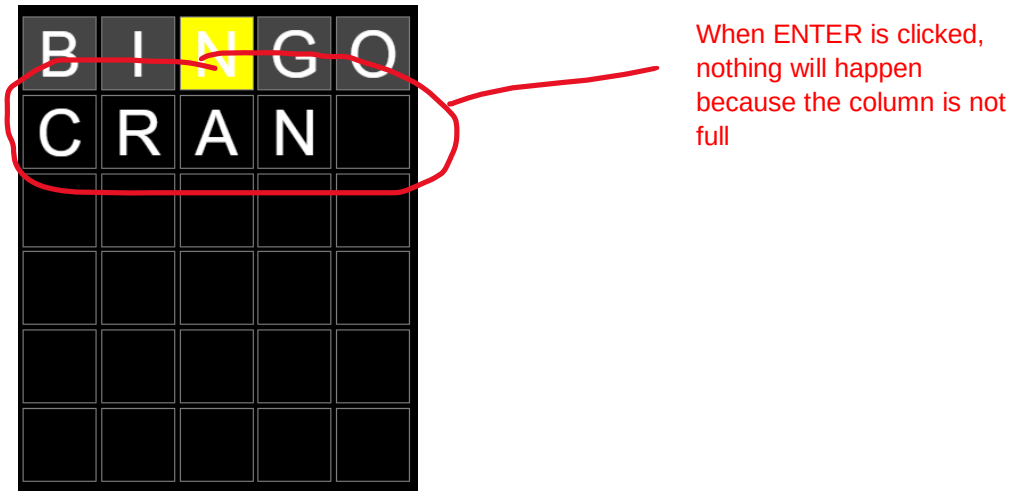
```
function removeLetter(){



}
```

☐ Write the *checkWord* function

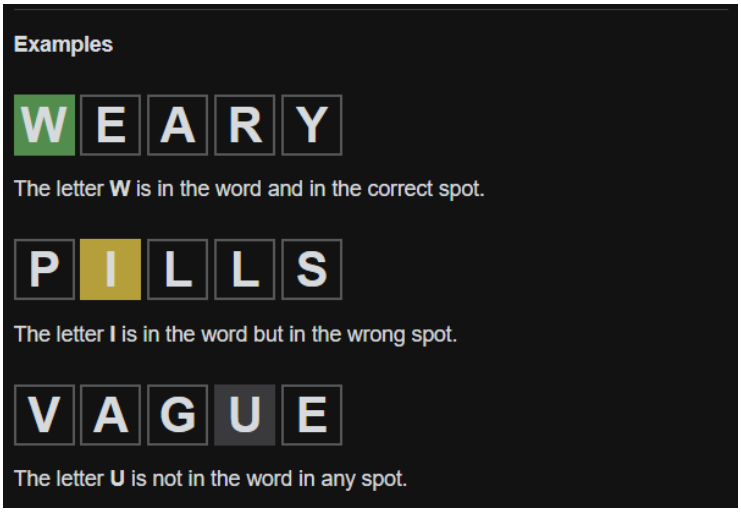
When the ENTER button is clicked, the *checkWord* function is called.

checkWord will only execute if a column is complete,



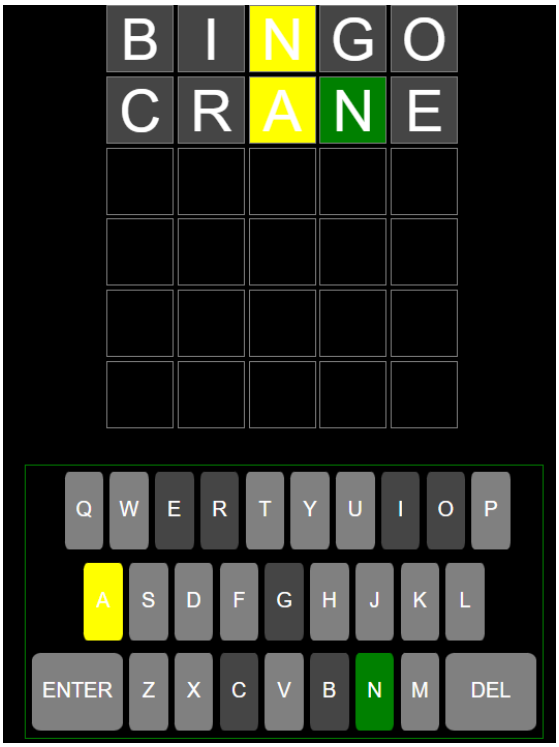


checkWord evaluates the word using the rules illustrated below.



If a letter is found and in the correct location in the word, the corresponding letterButton will turn green. If the letter is found and in the wrong spot, the corresponding letterButton will turn yellow. If the letter is not in the word, the letterButton will turn gray.

This behavior is illustrated below. The word to guess is SAUNA.



Each time the ENTER button is clicked *checkWord* is called. *checkWord* should only check the word if the last column is complete. You can keep track of which column is complete with a *letterCount* variable mentioned previously. For each letterBox in the column to be checked you must evaluate whether (1) the letter is in the word and in the correct location (2) the letter is in the word and in the wrong location (3) the letter is not in the word. For each case, you must also style the background color of the letterBox and the corresponding letterButton the correct color. Feel free to declare additional global variables as needed.

The *includes* function from the JavaScript String library is useful for completing this task. Below is an illustration of how to use *includes*.

```
var text = "Hello world, welcome to the universe.";
var result = text.includes("world");//returns true
```

□ Receive Credit for this lab guide

Submit this portion of the lab to Pluska to receive credit for the lab guide. Once received, your completed code challenges will also be graded and will count towards your final lab grade.