Hang person game

By the end of this task set you should have a simple hang-person game working

PRIOR UNDERSTANDING

- Loading a text file into a list/array
- Using random number generator

RESOURCES

Get the list of words to use in your game from `https://github.com/paulbaumgarten/data-sets`, it is a file called `hangperson-words.txt`.

TASK ONE

Create a function that can be used to hiding the letters not yet guessed of a word. The function should receive the `special_word` and `letters_guessed` as parameters, and then return the modified version that hides letters not guessed.

Example special word	Example letters guessed	Example return value
`"secret"`	`["a","b","c","d","e"]`	`"_e__e_"`
`"elephant"`	`["a","e","s","n","t"]`	`"e_e__ant"`

TASK TWO

Your hang-person game needs a number of "images" to draw, depicting how close the player is to game over. This can be done through multi line text strings. For instance, the final "image" could look like...

```
  +---+
  |   |
  O   |
 /|\  |
 / \  |
      |
=========
```

You should create a function that, based on the number provided through a parameter, will draw your hang-person at various stages of the game. You can decide how many chances to give your player.

TASK THREE

Create a function that

- Load the words text file into a list/array
- Use the random number generator to randomly select one item from the list as the secret word
- Returns the secret word

TASK FOUR

It's time to use the functions you created in tasks 1, 2 and 3 to make your functioning hang-person game.

REMEMBER

- Your program must include appropriate prompts for the entry of data.
- Error messages and other output need to be set out clearly.
- All variables, constants and other identifiers must have meaningful names.