Overview

This project is to be accomplished individually.

Using Python, write a simple text-based implementation of the popular word game, ***Hangman***.

Hangman is a game for two players. One player thinks of a word; and the other player tries to guess it. The word is initially represented by a row of dashes (one dash per letter). The guessing player successively suggests a letter that is believed to occur in the word.

If the suggested letter does occur in the word, the other player writes down the letter in all of its correct positions.

If the suggested letter does not occur in the word, the other player draws a part of a stick figure of a man hanged on gallows.

The game ends when the guessing player completes the word, or the other player completes the stick figure.

Simplifying Assumptions

To simplify the implementation, the following assumptions are given:

- Only capital letters will be used.
- The word to be guessed cannot contain spaces.
- The stick figure is **not** required. Displaying the number of remaining incorrect guesses will suffice.

Specifications

1. Save your Python script as **hangman.py**.

2. Your Python script file should contain header information in comments, containing your full names, your ID numbers, and the date you completed your program. It should also contain a comment block for the certification of authorship, directly following the header information. Follow this code certification template for project 1.

3. A scaffold file has been provided to serve as a guide for completing this project. You have the option to make use of this file either wholly or in part. You are NOT required to pattern your solution based on this scaffold. You may choose to define additional functions or remove existing ones. Or you may choose to ignore this completely and create your own implementation.

4. Place your Python script file in a folder named according to the convention: **Project1-IDNumber-Surname** (write surnames in alphabetical order)

5. In the folder created above, include a ***properly accomplished*** Certificate of Authorship named **Project1-COA-IDNumber-Surname.pdf**.
    a. Title of Submission: Project 1 - Hangman
    b. Type of Submission: Program

6. Archive your folder in zip format. Properly archiving it will produce a file named **Project1-IDNumber-Surname.zip**.

7. Submit it through the designated submission page by **Monday, 14 March 2022, 11:59 PM.**

Sample Simulations
Sample executions of a working program are shown below. Lines in **red** are user input. Lines in **blue** are comments, and are **NOT** part of the output. **Your program must follow the prompts and output format as seen in these simulations. This includes the characters, character spacing, line spacing, and the letter casing. Study these sample simulations carefully.**

```
LET'S PLAY HANGMAN!

Please enter a word for the other player to guess:
HELLO
(At this point, we just assume that we are able to successfully hide the word from the
other player.)

Guess the word, 6 guess(es) left: -----
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
E

Guess the word, 6 guess(es) left: -E---
Unused letters: ABCDFGHIJKLMNOPQRSTUVWXYZ
I

Guess the word, 5 guess(es) left: -E---
Unused letters: ABCDFGHJKLMNOPQRSTUVWXYZ
O

Guess the word, 5 guess(es) left: -E--O
Unused letters: ABCDFGHJKLMNPQRSTUVWXYZ
T

Guess the word, 4 guess(es) left: -E--O
Unused letters: ABCDFGHJKLMNPQRSUVWXYZ
S

Guess the word, 3 guess(es) left: -E--O
Unused letters: ABCDFGHJKLMNPQRUVWXYZ
R

Guess the word, 2 guess(es) left: -E--O
Unused letters: ABCDFGHJKLMNPQUVWXYZ
P

Guess the word, 1 guess(es) left: -E--O
Unused letters: ABCDFGHJKLMNQUVWXYZ
L

Guess the word, 1 guess(es) left: -ELLO
Unused letters: ABCDFGHJKMNQUVWXYZ
H

Guess the word, 1 guess(es) left: HELLO
Unused letters: ABCDFGJKMNQUVWXYZ
CONGRATULATIONS! YOU WIN!
```

Another sample execution of a working program is shown below.

```
LET'S PLAY HANGMAN!

Please enter a word for the other player to guess:
WEATHER

Guess the word, 6 guess(es) left: -------
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
A

Guess the word, 6 guess(es) left: --A----
Unused letters: BCDEFGHIJKLMNOPQRSTUVWXYZ
C

Guess the word, 5 guess(es) left: --A----
Unused letters: BDEFGHIJKLMNOPQRSTUVWXYZ
D

Guess the word, 4 guess(es) left: --A----
Unused letters: BEFGHIJKLMNOPQRSTUVWXYZ
E

Guess the word, 4 guess(es) left: -EA--E-
Unused letters: BFGHIJKLMNOPQRSTUVWXYZ
B

Guess the word, 3 guess(es) left: -EA--E-
Unused letters: FGHIJKLMNOPQRSTUVWXYZ
K

Guess the word, 2 guess(es) left: -EA--E-
Unused letters: FGHIJLMNOPQRSTUVWXYZ
Z

Guess the word, 1 guess(es) left: -EA--E-
Unused letters: FGHIJLMNOPQRSTUVWXY
T

Guess the word, 1 guess(es) left: -EAT-E-
Unused letters: FGHIJLMNOPQRSUVWXY
L

Guess the word, 0 guess(es) left: -EAT-E-
Unused letters: FGHIJMNOPQRSUVWXY
SORRY, YOU ARE HANGED!
```

Another sample execution of a working program is shown below.

```
LET'S PLAY HANGMAN!

Please enter a word for the other player to guess:
BANANA

Guess the word, 6 guess(es) left: ------
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
2

Please choose a valid letter. (Invalid input will NOT lead to guess deduction.)

Guess the word, 6 guess(es) left: ------
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
A

Guess the word, 6 guess(es) left: -A-A-A
Unused letters: BCDEFGHIJKLMNOPQRSTUVWXYZ
G

Guess the word, 5 guess(es) left: -A-A-A
Unused letters: BCDEFHIJKLMNOPQRSTUVWXYZ
V

Guess the word, 4 guess(es) left: -A-A-A
Unused letters: BCDEFHIJKLMNOPQRSTUWXYZ
G

You have already used that letter. (Reused letters will NOT lead to guess deduction.)

Guess the word, 4 guess(es) left: -A-A-A
Unused letters: BCDEFHIJKLMNOPQRSTUWXYZ
B

Guess the word, 4 guess(es) left: BA-A-A
Unused letters: CDEFHIJKLMNOPQRSTUWXYZ
N

Guess the word, 4 guess(es) left: BANANA
Unused letters: CDEFHIJKLMOPQRSTUWXYZ
CONGRATULATIONS! YOU WIN!
```

Hint: For invalid characters, you do not need to create a list of ALL the invalid characters. You can just compare the input against a list of all the allowable characters in the game.

Rubrics
Correct implementation of all the features will merit a highest possible grade of B (81-86), provided that:

- Project defense is successful
- Project is submitted on time and all submission instructions are followed

Successful implementation of any or all of these additional features will give students the chance to earn a higher grade:

- **Case-insensitive comparison** (e.g. if the word is `BANANA`, an input of `b` is considered correct)

- **Set the number of guesses allowed**

```
Please enter a word for the other player to guess:
LIGHT

Please enter the number of guesses allowed:
7

Guess the word, 7 guess(es) left: -----
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
T
```

- **Play again**

```
Guess the word, 1 guess(es) left: HELLO
Unused letters: ABCDFGJKMNQUVWXYZ
CONGRATULATIONS! YOU WIN!

Would you like to play again? (Yes/No)
Yes (If the player inputs No, the program ends.)

Please enter a word for the other player to guess:
GLASS

Please enter the number of guesses allowed:
5

Guess the word, 5 guess(es) left: -----
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Y
```

- **Program chooses a random word from a list**

```
LET'S PLAY HANGMAN!

Would you like the program to choose a random word? (Yes/No)
Yes (If the player inputs No, then ask for the word to be guessed.)

Here is the word to be guessed by your opponent: CHOCOLATE

Please enter the number of guesses allowed:
8

Guess the word, 8 guess(es) left: ---------
Unused letters: ABCDEFGHIJKLMNOPQRSTUVWXYZ
C
```

- **Display the Hangman Stick Figure**
  After each guess, display the state of the stick figure using only characters that can be displayed on the terminal. If you choose to implement this along with the ability to change the number of guesses, the stick figure must adjust correctly with respect to the chosen number.

Note that implementation of all these additional features will **NOT** guarantee a grade of A. The code will also be evaluated based on readability and minimization/elimination of redundancies. For the possibility of gaining more points, students may implement additional features as long as those are relevant to the game. Original features (meaning no other group has implemented the same) may earn higher points as well, depending on the level of difficulty. Also, note that these additional features will **NOT** serve as substitutes for the required features stated in this project specification sheet. You must still implement the required features in the manner described in this specification. If your additional features require the placement of additional prompts and outputs, you may do so, **provided that you do not alter the existing prompts and outputs for the required features**.

Note that aside from implementing the features (both required and additional) the students must also be able to clearly explain their source code. This will be done during the project defense. The score for this will account for a portion of the total project grade.

Project defense sign-up sheets will be released during the week of the submission deadline.

Restrictions
The following function/method from the Python Standard Library **MAY** be used:
- `random()` (from the `random` module)

NOTE: The `random` module has more functions other than `random()`, but you are **NOT** allowed to use those other functions (e.g. `randint(a,b)`, `choice(seq)`, etc...) to implement the random word add-on feature.

The following keywords, functions/methods are **NOT** allowed to be used:

`del`	`pop()`	`remove()`	`index()`
`join()`	`upper()`	`lower()`	`casefold()`
`swapcase()`	`replace()`	`chr()`	`ord()`

- `del` is used to delete objects
- `pop()` and `remove()` are used to remove items from a list.
- `index()` is used to retrieve the index of an item in a list
- `join()` is used to concatenate strings in a list
- `upper()`, `lower()`, `casefold()`, `swapcase()` are string methods that convert character casing
- `replace()` is used to replace a specified string with another specified string
- `chr()` returns a character based on a specified unicode point
- `ord()` returns a unicode point based on a specified character

Regarding the `in` operator:
- You **MAY** use the `in` operator in a `for` loop header (e.g. `for n in nums`)
- You may **NOT** use the `in` operator as a comparison operator (e.g. `'A' in ['A', 'B', 'C']`)

The `in` operator as a comparison operator returns `True` if the value being checked can be found in the sequence (e.g. `'A' in ['A', 'B', 'C']` returns `True`, but `'D' in ['A', 'B', 'C']` returns `False`).

Instead of using these keywords, functions/methods and operators, you are expected to create your own implementations of these relevant features using some combination of the following tools and constructs:

loops, conditionals, lists, and list comprehensions. Emphasis on **some combination**. You are **NOT** required to use all of them for every single feature.

If you wish to use other commands not clarified in this list, please email your instructors first before proceeding.

Rationale behind these restrictions:
Project 1 serves to assess your understanding of the fundamental topics discussed so far, but the use of the aforementioned keywords, functions/methods, and operators removes the rigor that is required in order to obtain a clear assessment, hence the restrictions. But outside of these course assessments, we encourage you to use these other tools and constructs in order to deepen your understanding of the Python language.

IMPORTANT REMINDERS:

- You must cite all of your sources in the Certificate of Authorship. This includes other individuals you have consulted with in any manner. You must explain in the citation the extent of the help you received from these individuals. Note that you are allowed to consult other sources only for clarification on the topics discussed in the modules, but **NOT** for solutions to this requirement.
- There is no need to implement a formal citation style. You simply need to indicate your sources clearly.
- During the project defense, students will be asked to explain how their code works. They will also be asked to perform some debugging and code tracing.
- There is no need to cite the slides and videos provided in this course.
- There is no need to cite your instructor(s) for this course.
- Double-check to see that you have successfully uploaded the correct file.