COMP 1039

Problem Solving and Programming

Programming Assignment 2

School of Computer and Information Science
The University of South Australia
May, 2019

Contents

Introduction

Assignment Overview

Graduate Qualities

Part I Specification

- Practical Requirements (Part I)
- Stages (Part I)

Part II Specification

- Practical Requirements (Part II)
- Stages (Part II)

Submission Details

Extensions and Late Submissions

Academic Misconduct

Marking Criteria

Character class explained – you should read this! :)

Sample Output – Part I

Sample Output – Part II

Useful Built-In Python Functions – required for part II

INTRODUCTION

This document describes the second assignment for Problem Solving and Programming.

The assignment is intended to provide you with the opportunity to put into practice what you have learnt in the course by applying your knowledge and skills to the implementation of a **Python module** (that contains functions that operate on lists) and **a program that will maintain information on characters (heroes and villains).**

This assignment is an **individual task** that will require an **individual submission**. **You will be required to submit your work via learnonline before Tuesday 11 June (week 13), 10:00 am (internal students). You will also be required to present your work to your practical supervisor during your practical session held in week 13 of the study period.** Important: You must attend the practical session that you have been attending all study period in order to have your assignment marked. External student will be required to submit their work via Learnonline before Friday 14 June (week 13), 11:30 pm.

This document is a kind of specification of the required end product that will be generated by implementing the assignment. Like many specifications, it is written in English and hence will contain some imperfectly specified parts. Please make sure you seek clarification if you are not clear on any aspect of this assignment.

ASSIGNMENT OVERVIEW

There are **two parts** to this assignment:

Part I: Writing a Python Module (list manipulation functions)

You are required to implement a Python module that contains functions that manipulate lists. Please ensure that you read sections titled 'Part I specification' below for further details.

Part II: Manage character (hero and villain) information

You are required to write a Python program that will manage character (heroes and villain) information. Character (hero and villain) information will be stored in a text file that will be read in when the program commences. Once the initial character (hero and villain) information has been read in from the file, the program should allow the user to interactively query and manipulate the character (hero and villain) information. Please ensure that you read sections titled 'Part II specification' below for further details.

Please ensure that you read sections titled 'Part I Specification' and 'Part II Specification' below for further details.

GRADUATE QUALITIES

By undertaking this assessment, you will progress in developing the qualities of a University of South Australia graduate.

The Graduate qualities being assessed by this assignment are:

- The ability to demonstrate and apply a body of knowledge (GQ1) gained from the lectures, workshops, practicals and readings. This is demonstrated in your ability to apply problem solving and programming theory to a practical situation.
- The development of skills required for lifelong learning (GQ2), by searching for information and learning to use and understand the resources provided (Python standard library, lectures, workshops, practical exercises, etc); in order to complete a programming exercise.
- The ability to effectively problem solve (GQ3) using Python to complete the programming problem. Effective problem solving is demonstrated by the ability to understand what is required, utilise the relevant information from lectures, workshops and practical work, write Python code, and evaluate the effectiveness of the code by testing it.
- The ability to work autonomously (GQ4) in order to complete the task.
- The use of communication skills (GQ6) by producing code that has been properly formatted; and writing adequate, concise and clear comments.
- The application of international standards (GQ7) by making sure your solution conforms to the standards presented in the Python Style Guide slides (available on the course website).

PART I SPECIFICATION – WRITING A PYTHON MODULE (LIST MANIPULATION FUNCTIONS)

You are required to write a `list_function.py` module (containing only the functions listed below). This file is provided for you (on the course website) however, you will need to modify this file by writing code that implements the functions listed below. **Please read the slides on modules available on the course website if you would like more information on modules.**

You are required to implement a Python module containing the following functions:

1. Write a function called **`length(my_list)`** that takes a list as a parameter and returns the length of the list. You must use a loop in your solution. You **must not** use built-in functions, list methods or string methods in your solution.

2. Write a function called **`to_string(my_list, sep=', ')`** that takes a list and a separator value as parameters and returns the **string** representation of the list (separated by the separator value) in the following form:

   ```
   item1, item2, item3, item4
   ```

   The separator value **must** be a default argument. i.e. sep=', '

   You must use a loop in your solution. You **must not** use built-in functions (other than the `range()` and `str()` functions), slice expressions, list methods or string methods in your solution. You may use the concatenation (+) operator to build the string. You **must** return a string from this function.

3. Write a function called **`find(my_list, value)`** that takes a list, and a value as parameters. The function searches for the value in the list and returns the index at which the first occurrence of value is found in the list. The function returns -1 if the value is not found in the list.

4. Write a function called **`insert_value(my_list, value, insert_position)`** that takes a list, a value and an insert_position as parameters. The function returns a **copy** of the list with the `value` inserted into the list (`my_list`) at the index specified by `insert_position`. Check for the `insert_position` value exceeding the list (`my_list`) bounds. If the `insert_position` is greater than the length of the list, insert the value at the end of the list. If the `insert_position` is less than or equal to zero, insert the value at the start of the list. You **must** use a loop(s) in your solution. You may make use of the *`list_name`*`.append(item)` method in order to build the new list. You **must not** use built-in functions (other than the `range()` function), slice expressions, list methods (other than the `append()` method) or string methods in your solution.

5. Write a function called **`remove_value(my_list, remove_position)`** that takes a list and a remove_position as parameters. The function returns a **copy** of the list with the item at the index specified by `remove_position`, removed from the list. Check for the `remove_position` value exceeding the list (`my_list`) bounds. If the `remove_position` is greater than the length of the list, remove the item at the end of the list. If the `remove_position` is less than or equal to zero, remove the item stored at the start of the list. You must use a loop in your solution. You may make use of the *`list_name`*`.append(item)` method in order to build the new list. You **must not** use built-in functions (other than the `range()` function), slice expressions, list methods (other than the `append()` method) or string methods in your solution.

6. Write a function called **`reverse(my_list, number=-1)`** that takes a list and a number as parameters. The function returns a **copy** of the list with the first `number` of items reversed. The `number` parameter **must** be a default argument. If the default argument for `number` is given in the function call, only the first number of items are reversed. If the default argument for `number` is not provided in the function call, then the entire list is reversed. Check for the number value exceeding the list bounds (i.e. is greater than the length of the list). If the number value exceeds the list bounds, then make the number value the length of the list (i.e. the entire list is reversed). If the number value entered is less than two, then return a **copy** of the list with no items reversed. You must use a loop(s) in your solution. You may make use of the *`list_name`*`.append(item)` method in order to build the new list. You **must not** use built-in functions (other than the `range()` function), slice expressions, list methods (other than the `append()` method) or string methods in your solution.

Reverse example:

```
a)
numList = [1, 2, 3, 4, 5, 6, 7]
number = 4
```

The call to `reverse(numList, number)` should return the new list `[4, 3, 2, 1, 5, 6, 7]`.

```
b)
numList = [1, 2, 3, 4, 5, 6, 7]
```

The call to `reverse(numList)` should return the new list `[7, 6, 5, 4, 3, 2, 1]`.

Please note:
You must test your functions to ensure that they are working correctly. **So you do not have to write your own test file, one has been provided for you.** The `assign2_partI_test_file.py` file is a test file that contains code that calls the functions contained in the `list_function.py` module. **Please do not modify the test file.**

PRACTICAL REQUIREMENTS (PART I)

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution WILL include the use of:

- The supplied `list_function.py` module (containing the functions listed below). This is provided for you – **you will need to modify this file**.
- Functions (`length`, `to_string`, `find`, `insert_value`, `remove_value` and `reverse`) implemented adhering to the assignment specifications.
- The supplied `assign2_partI_test_file.py` file. This is provided for you – **please DO NOT modify this file**.
- Well constructed `while` loops. (Marks will be lost if you use `break` statements or the like in order to exit from loops).
- Well constructed `for` loops. (Marks will be lost if you use `break` statements or the like in order to exit from loops).
- Appropriate `if/elif/else` statements.
- Output that **strictly** adheres to the assignment specifications.
- Good programming practice:
    - Consistent commenting and code layout. You are to provide comments to describe: your details, program description, all variable definitions, all functions, and every significant section of code.
    - Meaningful variable names.

- Your solutions **MAY** make use of the following:
    - Built-in functions `range()` and `str()`.
    - List method `append()` to create/build new lists. i.e. `*list_name*.append(item)`.
    - Concatenation (+) operator to create/build new strings.
    - Comparison operators (==, !=, <, >, etc).
    - Access the individual elements in a list with an index (one element only). i.e. `list_name[index]`.
    - Use of any of the functions you have written as part of the assignment. i.e. length() function.

- Your solutions **MUST NOT** use:
    - Built-in functions (other than `range()` and `str()` functions).
    - Slice expressions to select a range of elements from a list. i.e. `list_name[start:end]`.
    - List methods (other than the `append()` method. i.e. `*list_name*.append(item)`).
    - String methods.
    - **Do not** use `break`, or `continue` statements in your solution – doing so will result in a significant mark deduction. **Do not** use the `return` statement as a way to break out of loops. **Do not** use `quit()` or `exit()` functions as a way to break out of loops.

Please ensure that you use Python 3.7.2 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.7.2 (or latest version).

STAGES (PART I)

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1

You will need both the `list_function.py` and `assign2_partI_test_file.py` files for this assignment. **These have been provided for you.** Please **download both of these files from the course website** and ensure that they are in the same directory as each other.

Test to ensure that this is working correctly by opening and running the `assign2_partI_test_file.py` file. If this is working correctly, you should now see the following output in the Python shell when you run your program:

```
Start Testing!

length Test
In function length()
List length: None
In function length()
List length: None

to_string Test
In function to_string()
List is: None
In function to_string()
List is: None
In function to_string()
List is: None

find Test
In function find()
None
In function find()
None

insert_value Test
In function insert_value()
None
In function insert_value()
None
In function insert_value()
None
In function insert_value()
None

remove_value Test
In function remove_value()
None
In function remove_value()
None
In function remove_value()
None

reverse Test
In function reverse()
None
In function reverse()
None
In function reverse()
None

End Testing!
```

Stage 2

Implement one function at a time. The following implementation order is a recommendation only:

- length() – *you may find this function in the lecture slides… :)*
- to_string()
- find()
- remove_value()
- insert_value()
- reverse()

Place the code that implements each function in the appropriate place in the `list_function.py` file.

For example, if you were implementing the `length()` function, you would place the code that calculates and returns the length of the list under the comment 'Place your code here' (within the length function definition) seen below.

```
# Function length() – place your own comments here…  : )
def length(my_list):

    # This line will eventually be removed - used for development purposes only.
    print("In function length()")

    # Place your code here
```

Test your function by running the `assign2_partI_test_file.py` test file to ensure each function is working correctly before starting on the next function.

Compare your output with the sample output provided (at the end of this document) to ensure that your function is working as it should.

Stage 3

Finally, check the sample output (see section titled 'Sample Output – Part I' towards the end of this document) and if necessary, modify your functions so that:

- The output produced by your program **EXACTLY** adheres to the sample output provided.
- Your program behaves as described in these specs and the sample output provided.

PART II SPECIFICATION – MANAGE CHARACTER INFORMATION

Write a **menu driven program** called `part2_yourEmailId.py` that will allow the user to enter commands and process these commands until the quit command is entered. The program will store and maintain character information (using a List of Character objects). Character information will be stored in a text file that will be read in when the program commences. Once the initial character data has been read in from the file, the program should allow the user to interactively query and manipulate the character information.

Input

When your program begins, it will read in character (hero and villain) information from a file called `characters.txt`. This is a text file that stores information pertaining to characters (heroes and villains). An example input file called `characters.txt` can be found on the course website (under the Assessment tab). You may assume that all data is in the correct format. The name of the character (hero or villain) is stored on a separate line. The very next line contains the secret identity of the character (hero or villain). The very next line contains a character specifying whether the character is a hero ('h') or villain ('v'), followed by the number of battles fought, battles won, battles lost, battles drawn, and a health value. This information is stored on one line and is separated by the space character as seen in Figure 1 below:

After the program has stored the data (using a List of Character objects), it will enter interactive mode as described in the following section.

```
Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
```

Figure 1: *Character information file format (`characters.txt`).*

Interactive Mode

Your program should enter an interactive mode after the character (hero and villain) information has been read from the file. The program will allow the user to enter commands and process these commands until the quit command is entered. The following commands should be allowed:

1. **list:**
   Displays for all characters, the character's name, number of battles, battles, won, lost, drawn, and their health value. Outputs the contents of the character list (heroes and villains) as seen below in the section titled Screen Format. Please read the section at the end of this handout titled – 'Useful Built-In Python Functions – required for part II'.

2. **heroes:**
   Displays for all heroes, the character's name, number of battles, battles, won, lost, drawn, and their health value. Outputs the contents of the character list (heroes and villains) as seen below in the section titled Screen Format. Please read the section at the end of this handout titled – 'Useful Built-In Python Functions – required for part II'.

3. **villains:**
   Displays for all villains, the character's name, number of battles, battles, won, lost, drawn, and their health value. Outputs the contents of the character list (heroes and villains) as seen below in the section titled Screen Format. Please read the section at the end of this handout titled – 'Useful Built-In Python Functions – required for part II'.

4. **search:**
   Prompts for and reads the character's (hero/villain's) name and searches for the character in the character (hero and villain) list. If the character is found in the character list, the character's name, secret identity, battles fought, won, lost, drawn, and their health value, are displayed to the screen as seen below (in the section titled Screen Format). If the character is not found in the list of characters (heroes and villains), an error message stating the character has not been found is displayed to the screen.

5. **reset:**
   Prompts for and reads the character's (hero/villain's) name and searcher for the character in the character list (heroes and villains). If the character is found in the list of characters (heroes and villains), the character's health value is reset to 100.). If the character is not found in the list of characters (heroes and villains), an error message stating the character has not been found is displayed to the screen.

6. **add:**
   Prompts for and reads the character's (hero or villain's) name, secret identity and whether the character is a hero or villain. If the character does not already exist (i.e. a match is not found on the character's name), the character is added to the list of characters (heroes and villains). If the character is added, the health value is initialised to 100 and all other data members (number battles, no won, no lost, no drawn), are initialised to zero and a message is displayed to the screen indicating that this has been successfully added.

   The character information must be added after the last character entry stored in the list (i.e. at the end of the list). If the character is already stored in the character list, an error message is displayed. No duplicate entries are allowed.

7. **remove:**
   Prompts for the character's name. If the character is found, he/she is removed from the list of characters (heroes and villains) and a message is displayed to the screen indicating that this has been done. If the character is not found in the character list, an error message is displayed.

8. **battle:**
   Prompts for the name of the two opponents to do battle. The program searches for each character in the list of characters (heroes and villains) and if the character is not found in the list of characters, an error message is displayed to the screen and the user is asked to enter another character (hero/villain).

   If the opponents are found in the list of characters (heroes and villains), he/she is then able to do battle and the number of battle rounds the heroes/villains will undertake (a number between 1-5 inclusive) is prompted for and read. One battle may have many (1-5 inclusive) rounds. The heroes/villains battle until either an opponent dies (health status is reduced to zero) or the number of battle rounds have been completed. For each individual battle round, the hero/villain will sustain a certain amount of damage to their health rating. The amount of damage sustained from the battle will be a randomly generated value between 0–50 inclusive. After each round, each opponents damage value (i.e. randomly generated number between 0–50 inclusive) and current health value (i.e. calculated by: health value – damage value) are displayed to the screen.

   After every battle (however many rounds), a winner is determined (i.e. the opponent with the higher health value wins the battle) and the opponents' battle statistics are updated (i.e. number of battles, no won, no lost, etc…) accordingly.

9. **quit:**
   Causes the program to quit, outputting the contents of the character list (list of character objects) to a file (see section '*Final Output*' below for format).

Note:

The program should display an appropriate message if a character is not found matching a search criteria. Appropriate messages should also be displayed to indicate whether a command has been successfully completed.

Please refer to the sample output (at the end of this handout) to ensure that your program is behaving correctly and that you have the correct output messages.

Each time your program prompts for a command, it should display the list of available commands. See the sample output (at the end of this handout) to ensure that you have the output format correct.

For example:

```
Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]:
```

Menu input should be validated with an appropriate message being displayed if incorrect input is entered.

Screen Format

The **`list, heroes and villains`** commands (`display_characters()` function) should display the character (hero and villain) information in the following format:

```
===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================
```

The **`search`** command should display individual character (hero/villain) information to the screen in the following format:

```
All about Catwoman --> VILLAIN

Secret identity: Selina Kyle

Battles fought: 12
  > No won:   0
  > No lost:  6
  > No drawn: 6

Current health:  50%
```

Or if a hero…

```
All about Aquaman --> HERO

Secret identity: Arthur Curry

Battles fought: 8
  > No won:   2
  > No lost:  2
  > No drawn: 4

Current health:  30%
```

Final Output

When your program finishes (because you entered the quit command) your program should output the contents of the list of characters to a file called `new_characters.txt`.

The format of this file should **exactly** match that of the input file.

PRACTICAL REQUIREMENTS (PART II)

It is recommended that you develop this part of the assignment in the suggested stages.

It is expected that your solution WILL include the use of:

- Your solution in a file called `part2_`*`yourEmailId`*`.py`.
- The supplied `character.py` module (that defines the `Character` class). This is provided for you – **do NOT modify this file**.
- `Character` objects and methods (as appropriate) from the `Character` class definition. You are not required to implement the `Character` class. This is provided for you (character.py module) – **do NOT modify this file**.
- Appropriate and well constructed `while` and/or `for` loops. (Marks will be lost if you use `break` statements or the like in order to exit from loops).
- You **must** implement **each** function listed below (**or** you may call the appropriate function(s) defined in the `list_function` module (that you wrote in part I of this assignment)).
- Appropriate `if, if-else, if-elif-else` statements (as necessary).
- The following functions:
  - read_file(filename)

    This function takes a file name and reads the contents of that file into a list called character_list. The function returns the newly created list of character objects. You **must** use a loop in your solution. You **may** use String and/or List methods **in this function only**. You may find the String methods `split()` and `strip()` useful here. Please note: This function will be provided for you and explained during week 11's lecture. You may use your own function or use the function provided during week 11 lecture… the decision is yours. :)
  - write_to_file(filename, character_list)

    This function will output the contents of the character list (list of character objects) to a file in the same format as the input file. The file will need to be opened for writing in this function (and of course closed once all writing has been done). The function accepts the filename of the file to write to and the list of character objects. You **must** use a loop in your solution.
  - display_characters(character_list, display_type)

    This function will take the list of character objects and a display_type as a parameter and will output the contents of the list to the screen. If the display type is 0, all characters will be displayed to the screen. If the display type is 1, only heroes will be displayed to the screen. If the display type is 2, only villains will be displayed to the screen. This function displays the information to the screen in the format specified in the assignment specifications under the section - 'Screen Format'. You **must** use a loop in your solution. Please have a read of the section at the end of this handout titled – 'Useful Built-In Python Functions – required for part II'.
  - find_character(character_list, name)

    This function will take the character's name as input along with the list of character objects (character_list) and will return the position (index) of the character found in the character_list. If the character is not found, the function returns -1. You **must** use a loop in your solution. You **must not** use list methods in your solution.

- add_character(character_list, name, secret_identity, is_hero)

  This function takes the list of character objects and the character's (to be added) name, secret_identity and is_hero (True or False) as parameters. If the character already exists in the list of characters, display an error message to the screen. If the character does not exist in the characters (heroes and villains) list, the character is added. Create a new character object with the information read in (i.e. name, secret_identity and is_hero) and add the character object to the end of the list of characters. If the character is added, health is initialised to 100 and all other data members (number of battles, no won, no lost, no drawn), are initialised to zero and a message is displayed to the screen indicating this has been successfully added. This function returns the list of characters. You may use the `list_name.append(item)` method to add the new character object to the list of characters. You **must** call function `find_character()` from this function.

- remove_character(character_list, name)

  This function takes the list of character objects and the character's (to be removed) name as parameters. If the character is not found in the list of characters, display an error message to the screen. If the character does exist in the characters list, this function removes the character object and displays a message to the screen indicating that the character has been removed from the characters list. This function returns the list of characters. You may use the `list_name.append(item)` method in this function. You **must** call function `find_character()` from this function.

- do_battle(character_list, opponent1_pos, opponent2_pos)

  This function takes the list of characters and the position of the two characters that are about to do battle (i.e. position is the location/index of the character stored in the list of characters. This is useful so we can update the character's health value after every battle and battle statistics).

  This function prompts for and reads the number of battle rounds the heroes/villains will undertake (a number between 1-5 inclusive). The function allows the heroes/villains to battle until either an opponent dies (health status is reduced to zero) or the number of battle rounds have been completed. For each individual battle round, the hero/villain will sustain a certain amount of damage to their health rating. The amount of damage sustained from the battle will be a randomly generated value between (0 - 50 inclusive).

  General algorithm is as follows:

```
while (number of battle rounds have not been completed and both opponents are still alive)
    randomly generate a damage value sustained from battle and update opponent 1's health value by
        calling method update_health(damage1).
    randomly generate a damage value sustained from battle and update opponent 2's health value by
        calling method update_health(damage2).
    display opponent 1 round results (as seen in the sample output)
    display opponent 2 round results (as seen in the sample output)

determine the winner of the battles - the character with the most health left at the end of all the battle rounds is
the winner.

display the winner to the screen and also report if a character has died as a result of battle (refer to sample
output for layout).

update opponent 1's battle statistics.  Hint: Increment the number of battles by one, number won, lost, drawn
accordingly (call the appropriate methods to do this).  We increment battles by one because battles may consist
of a number of battle rounds.
```

update opponent 2's battle statistics. Hint: Increment the number of battles by one, number won, lost, drawn accordingly (call the appropriate methods to do this). We increment battles by one because battles may consist of a number of battle rounds.

Hint: this function MUST make use of the update_health(…) method as well as other methods that get character information and update the character's battle statistics, etc.

- Good programming practice:
  - Consistent commenting, layout and indentation. You are to provide comments to describe: your details, program description, **all** variable definitions, **all** function definitions and every significant section of code.
  - Meaningful variable names.

Your solutions **MAY** make use of the following:

- Built-in functions `int()`, `input()`, `print()`, `range()`, `open()`, `close()`, `len()`, `format()` and `str()`.
- Concatenation (+) operator to create/build new strings.
- The `list_name.append(item)` method to update/create lists.
- Access the individual elements in a string with an index (one element only). i.e. `string_name[index]`.
- Access the individual elements in a list with an index (one element only). i.e. `list_name[index]`.
- `Character` objects and methods (as appropriate).
- The `list_function.py` module (that you wrote in part I of this assignment). You may like to make use of some of the functions defined in the `list_function.py` module for this part of the assignment (as appropriate). Not all will be suitable or appropriate.

Your solutions **MUST NOT** use:

- Built-in functions (other than the `int()`, `input()`, `print()`, `range()`, `open()`, `close()` `len()`, `format()` and `str()` functions).
- Slice expressions to select a range of elements from a string or list. i.e. `name[start:end]`.
- String or list methods (i.e. other than those mentioned in the 'MAY make use' of section above).
- Global variables as described in week 8 lecture.
- **Do not** use `break`, or `continue` statements in your solution – doing so will result in a significant mark deduction. **Do not** use the `return` statement as a way to break out of loops. **Do not** use `quit()` or `exit()` functions as a way to break out of loops.

PLEASE NOTE: You are reminded that you should ensure that all input and output conform to the assignment specifications. If you are not sure about these details you should check with the sample output provided at the end of this document or post a message to the discussion forum in order to seek clarification.

Please ensure that you use Python 3.7.2 or a later version (i.e. the latest version) in order to complete your assignments. Your programs **MUST** run using Python 3.7.2 (or latest version).

STAGES (PART II)

It is recommended that you develop this part of the assignment in the suggested stages. Many problems in later stages are due to errors in early stages. **Make sure you have finished and thoroughly tested each stage before continuing.**

The following stages of development are recommended:

Stage 1
To begin, download the provided files (available on the course website called `part2_emailid.py`, and `character.py`) and re-name part2_emailid.py `part2_`*`yourEmailId`*`.py`. Define an empty list to store the character information. For example:

```
character_list = []
```

Stage 2
Create a character object from class `Character` (provided for you in the `character.py` module). Note: **you may simply download it from the course website under the Assessment tab. The `character.py` module should be placed in the same directory *`as part2_yourEmailId.py`* file.** Please do **NOT** modify the `character.py` module.

Import the `character` module and create a character object as seen below. You may wish to read the material on modules posted on the course website (under the assessment tab). You should also read the section titled 'Description of `character.py` module' included in this document.

```
character_list = []


new_character = character.Character("Deadpool", "Wade Wilson", True)
print(new_character)

character_list.append(new_character)
```

Make sure the program runs correctly. Once you have that working, back up your program. *Note: When developing software, you should always have fixed points in your development where you know your software is bug free and runs correctly.*

Once you are sure that your program is working correctly, you can comment out or remove the above statements – we just wanted to make sure it was working correctly before continuing! You can bring it back later (appropriately positioned if necessary…).

Stage 3
Write the code for functions `read_file()` and `display_characters()` as specified above. Add code to call these two functions to ensure they are working correctly. You may write your own `read_file()` function or you may use the one provided for you. The read_file() function will be explained in week 11 lecture.

Stage 4

Now that you know the information is being correctly stored in your characters list, write the code for function `write_to_file()`. Add code to call this function to ensure it is working correctly.

Stage 5

Implement the interactive mode, i.e. to prompt for and read menu commands. Set up a loop to obtain and process commands. Test to ensure that this is working correctly before moving onto the next stage. You do not need to call any functions at this point, you may simply display an appropriate message to the screen, for example:

Sample output:
```
Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: roger

Not a valid command - please try again.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

In list command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: heroes

In heroes command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: villains

In villains command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: search

In search command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: add

In add command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: remove

In remove command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: battle

In battle command


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit
```

Menu input should be validated with an appropriate message being displayed if incorrect input is entered by the user.

Stage 7

Implement one command at a time. Test to ensure the command is working correctly before starting the next command. Start with the `quit` and `list` commands as they do not need you to add anything further to the file other than ensuring that the function calls are in the correct place.

You should be able to see that for *most* commands there is a corresponding function(s).

For the remaining commands, the following implementation order is suggested (note: this is a guide only):

- `list` command (`display_characters()` function).
- `heroes` command (`display_characters()` function).
- `villains` command (`display_characters()` function).
- `search` command (`find_character()` function).
- `reset` command (calls the `find_character()` function).
- `add` command (`add_character()` function).
- `remove` command (`remove_character()` function).
- `battle` command (`do_battle()` function).

Stage 8

Ensure that you have validated all user input with an appropriate message being displayed for incorrect input entered by the user. Add code to validate all user input. Hint: use a while loop to validate input.

Stage 9

Finally, check the sample output (see section titled 'Sample Output – Part II' towards the end of this document) and if necessary, modify your code so that:

- The output produced by your program **EXACTLY** adheres to the sample output provided.
- Your program behaves as described in these specs and the sample output provided.

SUBMISSION DETAILS

You are required to do the following in order to submit your work and have it marked:

- Internal students:
  - You are required to submit an electronic copy of your program via learnonline **before Tuesday 11 June (week 13), 10 am (internal students)**.
  - **Internal students are also required to demonstrate your assignment to your practical supervisor during your week 13 practical class for marking. The supervisor will mark your work using the marking criteria included in this document. You MUST attend the practical session that you have been attending all study period in order to have your assignment marked.**

  Assignments submitted to learnonline, but not demonstrated during your allocated practical session, will NOT be marked. Likewise, assignments that have been demonstrated during the practical session, but have not been submitted via learnonline, will NOT be marked. Assignments are submitted to learnonline in order to check for plagiarism.

- External students:

  If you are an **external student**, you are only required to submit an electronic copy of your program via learnonline before Friday 14 June (week 13), 11:30pm (external students only). External students are **not** required to demonstrate in person.

All students (internal and external) must follow the submission instructions below:

Ensure that your files are named correctly (as per instructions outlined in this document).

Ensure that the following two files are included in your submission:

- `list_function.py`
- `part2_yourEmailId.py`

For example (if your name is James Bond, your submission files would be as follows):

- `list_function.py, part2_bonjy007.py`

All files that you submit must include the following comments.

```
#
# File: fileName.py
# Author: your name
# Email Id: your email id
# Description: Assignment 2 – place assignment description here…
# This is my own work as defined by the University's
# Academic Misconduct policy.
#
```

Assignments that do not contain these details may not be marked.

You must submit your program **before the online due date** and demonstrate your work to your marker. You will also be required to demonstrate that you have correctly submitted your work to learnonline. Work that has not been correctly submitted to learnonline will not be marked.

It is expected that students will make copies of all assignments and be able to provide these if required.

EXTENSIONS AND LATE SUBMISSIONS

There will be **no** extensions/late submissions for this course without one of the following exceptions:

1. A medical certificate is provided that has the timing and duration of the illness and an opinion on how much the student's ability to perform has been compromised by the illness. **Please note** if this information is not provided the medical certificate WILL NOT BE ACCEPTED. Late assessment items will not be accepted unless a medical certificate is presented to the Course Coordinator. The certificate must be produced as soon as possible and must cover the dates during which the assessment was to be attempted. In the case where you have a valid medical certificate, the due date will be extended by the number of days stated on the certificate up to five working days.
2. A Learning and Teaching Unit councillor contacts the Course Coordinator on your behalf requesting an extension. Normally you would use this if you have events outside your control adversely affecting your course work.
3. Unexpected work commitments. In this case, you will need to attach a letter from your work supervisor with your application stating the impact on your ability to complete your assessment.
4. Military obligations with proof.

Applications for extensions must be lodged via learnonline before the due date of the assignment.

Note: Equipment failure, loss of data, 'Heavy work commitments' or late starting of the course are not sufficient grounds for an extension.

ACADEMIC MISCONDUCT

ACADEMIC MISCONDUCT

Students are reminded that they should be aware of the academic misconduct guidelines available from the University of South Australia website.

Deliberate academic misconduct such as plagiarism is subject to penalties. Information about Academic integrity can be found in Section 9 of the *Assessment policies and procedures manual* at:
http://www.unisa.edu.au/policies/manual/

MARKING CRITERIA

Please note that the following is a guide only and may be subject to change (see next page for breakdown).

Other possible deductions:

- *Programming style:* Things to watch for are poor or no commenting, poor variable names, etc.
- *Submitted incorrectly:* -10 marks if assignment is submitted incorrectly (i.e. not adhering to the specs).

University of South Australia

Assessment feedback

Problem Solving and Programming (COMP 1039)
Assignment 2 - Weighting: 15% - Due: sp2, Week 13, 2019

NAME:	MAX MARK	MARK	COMMENT
PRODUCES CORRECT RESULTS (OUTPUT) - PART I			

length Test [5 marks]
`List length: 7`
`List length: 0`
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

to string Test [5 marks]
`List is: r, i, n, g, i, n, g`
`List is: r-i-n-g-i-n-g`
`List is:`
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

find Test [5 marks]
`3`
`-1`
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

insert value Test [5 marks]
`['one', 'two', 'three', 'four', 'five']`
`['p', 'i', 't']`
`['s', 'p', 'i', 't']`
`['s', 'p', 'i', 't', 's']`
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

remove value Test [5 marks]
`['r', 'i', 'g']`
`['i', 'n', 'g']`
`['r', 'i', 'n']`
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

reverse Test [5 marks]
`['d', 'u', 'd', 'e']`
`['b', 'o', 'r', 'e', 'd']`
`['d', 'u', 'd', 'e']`
 	
`End Testing!`
	30 marks		☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output
			☐ -1 No or incorrect output

ADHERES TO SPECIFICATIONS (CODE) - PART I			
Function `length(my_list); return length of list` Function `to_string(my_list, sep=', '); return str` Function `find(my_list, value); return index or -1` Function `insert_value(my_list, value, position); return copy of list` Function `remove_value(my_list, position); return copy of list` Function `reverse(my_list, number=-1); return copy of list`			☐ -2 Not to specs or -5 not implemented ☐ -2 Not to specs or -5 not implemented ☐ -2 Not to specs or -5 not implemented ☐ -2 Not to specs or -5 not implemented ☐ -2 Not to specs or -5 not implemented ☐ -2 Not to specs or -5 not implemented
Well constructed loops. Appropriate if statements. No global variables.			☐ -1 No or incorrect loops ☐ -1 No or incorrect if statements ☐ -1 Use of global variables
Should **not use** the following: • Built-in functions (other than range() and str() functions) • Slice expressions to select range of elements • String or list methods (other than list_name.append() method).			☐ -2 use of built-in functions not allowed ☐ -2 use of slicing i.e. [::-1] ☐ -2 use of methods not allowed
Use of `list_function.py` file. Use of `assign2_partI_test_file.py` file.			☐ -1 No or incorrect use of file ☐ -1 No or incorrect use of file ☐ -2 Using break/return to exit loops

PRODUCES CORRECT RESULTS (OUTPUT) - PART II
 	
`Please enter choice`
`[list, heroes, villains, search, reset, add, remove, battle, quit]:`
			☐ -2 No or incorrect line spacing
			☐ -2 No or incorrect menu display

list, heroes, villains `[4 marks]`
`==================================================`
`-     Character (heroes and villains) Summary     -`
`==================================================`
`-                             P  W  L  D  Health  -`
`--------------------------------------------------`
`-  Wonder Woman               5  5  0  0      90  -`
`--------------------------------------------------`
`-  Batman                     6  2  0  4      80  -`
`--------------------------------------------------`
`   :`
`--------------------------------------------------`
`-  Hulk                       7  2  1  4      80  -`
`--------------------------------------------------`
`-  Thanos                    10  2  0  8      90  -`
`--------------------------------------------------`
`==================================================`
			☐ -1 For each missing or incorrect info (up to 2 marks)
			☐ -1 For each formatting error (up to 2 marks)
			☐ -2 For no or incorrect heroes listing
			☐ -2 For no or incorrect villains listing

search `[4 marks]`
`All about Catwoman --> VILLAIN`
 	
`Secret identity: Selina Kyle`
 	
`Battles fought: 12`
`  > No won:   0`
`  > No lost:  6`
`  > No drawn: 6`
 	
`Current health:  50%`
			☐ -1 For each output/prompt/msg not to specs (up to 4 marks)

reset `[2 marks]`
`Please enter name: Batman`
 	
`Successfully updated Batman's health to 100`
			☐ -1 For each output/prompt/msg not to specs (up to 2 marks)

add `[4 marks]`
`Please enter name: Deadpool`
`Please enter secret_identity: Wade Wilson`
`Is this character a hero or a villain [h|v]? h`
 	
`Successfully added Deadpool to character list.`
	30 marks		☐ -1 For each output/prompt/msg not to specs (up to 4 marks)

remove `[4 marks]`
`Please enter name: Hulk`
 	
`Successfully removed Hulk from character list.`
			☐ -1 For each output/prompt/msg not to specs (up to 4 marks)

battle `[8 marks]`
`-- Battle --`
 	
`Wonder Woman versus Aquaman - 1 rounds`
 	
`Round: 1`
`  > Wonder Woman - Damage: 7 - Current health: 83`
`  > Aquaman - Damage: 7 - Current health: 23`
 	
`-- End of battle --`
 	
`** Wonder Woman wins! **`
			☐ -1 For each output/prompt/msg not to specs (up to 8 marks)
			Check to see whether stats and health are updating correctly as a result of battles. -2 if not updating correctly.

Output file (`new_characters.txt`) `[4 marks]`			☐ -2 If output file does not exist ☐ -1 If incorrect results in file ☐ -1 If output not to specs in file
ADHERES TO SPECIFICATIONS (CODE) - PART II While loop for menu/prompt (choice != "quit") Appropriate control structures (in general) Use of following functions: ▪ read_file(filename); return list of character objects ▪ display_characters(character_list, display_type) ▪ write_to_file(filename, character_list) ▪ find_character(character_list, name); return index or -1 ▪ add_character(character_list, name, secret_id, is_hero); return list of character objects; calls function find_character() ▪ remove_character(character_list, name); return list of character objects; calls function find_character() ▪ do_battle(character_list, opponent1_pos, opponent2_pos) Use of `character.py` file and `Character` class Should **not use** the following: Built-in functions, Slice expressions to select range of elements, String or list methods (other than list_name.append()), Global variables. *Validation of user input - messages:* `Not a valid command - please try again.`			☐ -2 No or incorrect loop ☐ -2 No or incorrect control structures ☐ -2 No or incorrect function read_file ☐ -2 No or incorrect function display_characters ☐ -2 No or incorrect function write_to_file ☐ -2 No or incorrect function find_character ☐ -2 No or incorrect function add_character ☐ -2 No or incorrect function remove_character ☐ -2 No or incorrect function do_battle ☐ -2 No or incorrect use of file ☐ -2 for **each** should not be using ☐ -2 No validation of user input ☐ -2 For using break/return/exit statements to exit loops
STYLE (BOTH PARTS): Comments (your details, prog. description, all variable definitions, functions & code), meaningful variable names.	5 marks		☐ -4 Insufficient comments ☐ -4 Inconsistent code layout ☐ -4 Non-descriptive variable names
TOTAL	**75 MARKS**		

DESCRIPTION OF `character.py` MODULE

The `character.py` module provides a character class definition called `Character`. This class definition is to be used to create character objects and is designed to be used for assignment 2 (part II) work. `Character` objects will be used to store character (hero and villain) information.

To make use of `class Character` defined within the `character` module, you will need to import the `character` module as seen below:

```
import character
```

To create a `Character` object, with name, secret_identity, hero status (and all other battle statistics; number of battles fought, no won, no lost, and no drawn initialised to zero and health value initialised to 100) you will need to do the following:

```
new_character = character.Character("Captain Marvel", "Carol Danvers", True)
```

The above code constructs a `Character` object storing character information about Captain Marvel.

`Character(name, secret_identity, hero_status)`	Constructs a `Character` object with name, secret identity, hero status (True / False) data members; all other data members are initialised to zero and health value initialised to 100.

`Character` Object Methods:

set _name(name)	Sets the character’s name.
get_name()	Returns the character’s name.
set _secret_identity(secret_identity)	Sets the character’s secret identity.
get_secret_identity()	Returns the character’s secret identity.
set _is_hero(hero)	Sets whether the character is a hero (Boolean True / False value).
get_is_hero()	Returns the is hero value (Boolean True or False).
set_no_battles(battles)	Sets the number of battles the character has fought in.
get_no_battles()	Returns the number of battles the character has fought in.
increment_no_battles(battles)	Increments the number of battles character has fought by battles (default 1).
set_no_won(won)	Sets the number of games character has won.
get_no_won()	Returns the number of games character has won.
increment_no_won(won)	Increments the number of games character has won by won (default 1).
set_no_lost(lost)	Sets the number of games character has lost.
get_no_lost()	Returns the number of games character has lost.
increment_no_lost(lost)	Increments the number of games character has lost by lost (default 1).
set_no_drawn(drawn)	Sets the number of games character has drawn.
get_no_drawn()	Returns the number of games character has drawn.
increment_no_drawn(drawn)	Increments the number of games character has drawn by drawn (default 1).
set_health(health)	Sets the character’s health value.
get_health()	Returns the character’s health value.
update_health(damage)	Updates the health value resulting from battles fought (by health - damage).
__str__()	Returns a string representation of the `Character` object.

A simple example:

You can construct a `Character` object and call it's methods as seen in the following example:

```
import character


# Create Character object with name "Captain Marvel"
new_character1 = character.Character("Captain Marvel", "Carol Danvers", True)

# Display the Character object to the screen as specified by the __str__ method.
print(new_character1)

# Update the number of battles fought by one
new_character1.increment_no_battles()

# Display the Character object to the screen as specified by the __str__ method.
print(new_character1)

# Create another Character object
new_character2 = character.Character("Spider-Man", "Peter Parker", True)

# Update character's health value
new_character2.update_health(20)

# Display the Character object to the screen as specified by the __str__ method.
print(new_character2)

# Create yet another Character object
new_character3 = character.Character("Catwoman", "Selina Kyle", False)

# Update total score of character
new_character3.update_health(30)

# Display the Character object to the screen as specified by the __str__ method.
print(new_character3)


# Create list of characters (Character objects)
character_list = []

# Append character objects to list
character_list.append(new_character1)
character_list.append(new_character2)
character_list.append(new_character3)

print("\n\nDisplaying character_list:")

# Iterate over character list and display to screen
for character in character_list:
    print(character.get_name(), character.get_secret_identity(), character.get_no_battles(),
character.get_health())
```

Output:

```
Captain Marvel             0  0  0  0     100
Captain Marvel             1  0  0  0     100
Spider-Man                 0  0  0  0      80
Catwoman                   0  0  0  0      70


Displaying character_list:
Captain Marvel Carol Danvers 1 100
Spider-Man Peter Parker 0 80
Catwoman Selina Kyle 0 70
```

SAMPLE OUTPUT – PART I

Sample output 1:

```
Start Testing!

length Test
List length: 7
List length: 0

to_string Test
List is: r, i, n, g, i, n, g
List is: r-i-n-g-i-n-g
List is: 

find Test
3
-1

insert_value Test
['one', 'two', 'three', 'four', 'five']
['p', 'i', 't']
['s', 'p', 'i', 't']
['s', 'p', 'i', 't', 's']

remove_value Test
['r', 'i', 'g']
['i', 'n', 'g']
['r', 'i', 'n']

reverse Test
['d', 'u', 'd', 'e']
['b', 'o', 'r', 'e', 'd']
['d', 'u', 'd', 'e']

End Testing!
```

SAMPLE OUTPUT – PART II

Sample output 1:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: display

Not a valid command - please try again.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: heroes

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: villains

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --
```

```
NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
```

Sample output 2:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: search

Please enter name: Deadpool

Deadpool is not found in character (heroes and villains) list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: search

Please enter name: Aquaman

All about Aquaman --> HERO

Secret identity: Arthur Curry

Battles fought: 8
  > No won:   2
  > No lost:  2
  > No drawn: 4

Current health:  30%


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit
```

```
-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
```

Sample output 3:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: reset

Please enter name: Captain Marvel

Captain Marvel is not found in character (heroes and villains) list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: reset

Please enter name: Iron Man

Successfully updated Iron Man's health to 100


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                              P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman                5  5  0  0      90  -
---------------------------------------------------
-  Batman                      6  2  0  4      80  -
---------------------------------------------------
-  The Joker                   5  1  0  4      80  -
---------------------------------------------------
-  Superman                    7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                   12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                     8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                   10  8  2  0     100  -
---------------------------------------------------
-  Hulk                        7  2  1  4      80  -
---------------------------------------------------
-  Thanos                     10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 100
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
```

Sample output 4:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                              P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman                5  5  0  0      90  -
---------------------------------------------------
-  Batman                      6  2  0  4      80  -
---------------------------------------------------
-  The Joker                   5  1  0  4      80  -
---------------------------------------------------
-  Superman                    7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                   12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                     8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                   10  8  2  0      50  -
---------------------------------------------------
-  Hulk                        7  2  1  4      80  -
---------------------------------------------------
-  Thanos                     10  2  0  8      90  -
---------------------------------------------------
```

```
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: add

Please enter name: Deadpool
Please enter secret_identity: Wade Wilson
Is this character a hero or a villain [h|v]? h

Successfully added Deadpool to character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: add

Please enter name: The Joker
Please enter secret_identity: Jack Napier
Is this character a hero or a villain [h|v]? v

The Joker already exists in character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
-  Deadpool                   0  0  0  0     100  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
Deadpool
Wade Wilson
h 0 0 0 0 100
```

Sample output 5:
```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: remove

Please enter name: Black Widow

Black Widow is not found in characters.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: remove

Please enter name: Wonder Woman

Successfully removed Wonder Woman from character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
```

```
---------------------------------------------------
-  Catwoman                12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                  8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                10  8  2  0      50  -
---------------------------------------------------
-  Hulk                     7  2  1  4      80  -
---------------------------------------------------
-  Thanos                  10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: remove

Please enter name: Thanos

Successfully removed Thanos from character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                           P  W  L  D  Health  -
---------------------------------------------------
-  Batman                   6  2  0  4      80  -
---------------------------------------------------
-  The Joker                5  1  0  4      80  -
---------------------------------------------------
-  Superman                 7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                  8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                10  8  2  0      50  -
---------------------------------------------------
-  Hulk                     7  2  1  4      80  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: remove

Please enter name: Catwoman

Successfully removed Catwoman from character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                           P  W  L  D  Health  -
---------------------------------------------------
-  Batman                   6  2  0  4      80  -
---------------------------------------------------
-  The Joker                5  1  0  4      80  -
---------------------------------------------------
-  Superman                 7  4  0  3     100  -
---------------------------------------------------
-  Aquaman                  8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                10  8  2  0      50  -
---------------------------------------------------
-  Hulk                     7  2  1  4      80  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: add

Please enter name: The Flash
Please enter secret_identity: Barry Allen
Is this character a hero or a villain [h|v]? h

Successfully added The Flash to character list.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                           P  W  L  D  Health  -
---------------------------------------------------
-  Batman                   6  2  0  4      80  -
```

```
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  The Flash                  0  0  0  0     100  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
The Flash
Barry Allen
h 0 0 0 0 100
```

Sample output 6:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: battle

Please enter opponent one's name: The Flash
The Flash is not found in character list - please enter another opponent!

Please enter opponent one's name: Wonder Woman
Please enter opponent two's name: Spider-Man
Spider-Man is not found in character list - please enter another opponent!

Please enter opponent one's name: Aquaman
Please enter number of battle rounds: 9
Must be between 1-5 inclusive.

Please enter number of battle rounds: -1
Must be between 1-5 inclusive.

Please enter number of battle rounds: 3


-- Battle --

Wonder Woman versus Aquaman - 3 rounds

Round: 1
  > Wonder Woman - Damage: 23 - Current health: 67
  > Aquaman - Damage: 50 - Current health: 0

-- End of battle --

-- Aquaman has died!  :(

** Wonder Woman wins! **


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               6  6  0  0      67  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    9  2  3  4       0  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 6 6 0 0 67
Batman
Bruce Wayne
h 6 2 0 4 80
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 7 4 0 3 100
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 9 2 3 4 0
Iron Man
Tony Stark
h 10 8 2 0 50
Hulk
Bruce Banner
h 7 2 1 4 80
Thanos
n/a
v 10 2 0 8 90
```

Sample output 7:

```
File     : wayby001_character_info.py
Author   : Batman
Stud ID  : 0123456X
```

```
Email ID : wayby001
This is my own work as defined by the University's Academic Misconduct Policy.


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     6  2  0  4      80  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   7  4  0  3     100  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: battle

Please enter opponent one's name: Superman
Please enter opponent two's name: Batman
Please enter number of battle rounds: 5


-- Battle --

Superman versus Batman - 5 rounds

Round: 1
  > Superman - Damage: 45 - Current health: 55
  > Batman - Damage: 1 - Current health: 79

Round: 2
  > Superman - Damage: 12 - Current health: 43
  > Batman - Damage: 14 - Current health: 65

Round: 3
  > Superman - Damage: 44 - Current health: 0
  > Batman - Damage: 3 - Current health: 62

-- End of battle --

-- Superman has died!  :(

** Batman wins! **


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     7  3  0  4      62  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   8  4  1  3       0  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  10  8  2  0      50  -
---------------------------------------------------
-  Hulk                       7  2  1  4      80  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: battle
```

```
Please enter opponent one's name: Iron Man
Please enter opponent two's name: Hulk
Please enter number of battle rounds: 2


-- Battle --

Iron Man versus Hulk - 2 rounds

Round: 1
  > Iron Man - Damage: 2 - Current health: 48
  > Hulk - Damage: 5 - Current health: 75

Round: 2
  > Iron Man - Damage: 49 - Current health: 0
  > Hulk - Damage: 11 - Current health: 64

-- End of battle --

-- Iron Man has died!  :(

** Hulk wins! **


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     7  3  0  4      62  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   8  4  1  3       0  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  11  8  3  0       0  -
---------------------------------------------------
-  Hulk                       8  3  1  4      64  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: reset

Please enter name: Hulk

Successfully updated Hulk's health to 100


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: reset

Please enter name: Iron Man

Successfully updated Iron Man's health to 100


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     7  3  0  4      62  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   8  4  1  3       0  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  11  8  3  0     100  -
---------------------------------------------------
-  Hulk                       8  3  1  4     100  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
```

```
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: battle

Please enter opponent one's name: Iron Man
Please enter opponent two's name: Hulk
Please enter number of battle rounds: 2


-- Battle --

Iron Man versus Hulk - 2 rounds

Round: 1
  > Iron Man - Damage: 38 - Current health: 62
  > Hulk - Damage: 14 - Current health: 86

Round: 2
  > Iron Man - Damage: 39 - Current health: 23
  > Hulk - Damage: 31 - Current health: 55

-- End of battle --

** Hulk wins! **


Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: list

===================================================
-     Character (heroes and villains) Summary     -
===================================================
-                             P  W  L  D  Health  -
---------------------------------------------------
-  Wonder Woman               5  5  0  0      90  -
---------------------------------------------------
-  Batman                     7  3  0  4      62  -
---------------------------------------------------
-  The Joker                  5  1  0  4      80  -
---------------------------------------------------
-  Superman                   8  4  1  3       0  -
---------------------------------------------------
-  Catwoman                  12  0  6  6      50  -
---------------------------------------------------
-  Aquaman                    8  2  2  4      30  -
---------------------------------------------------
-  Iron Man                  12  8  4  0      23  -
---------------------------------------------------
-  Hulk                       9  4  1  4      55  -
---------------------------------------------------
-  Thanos                    10  2  0  8      90  -
---------------------------------------------------
===================================================

Please enter choice
[list, heroes, villains, search, reset, add, remove, battle, quit]: quit


-- Program terminating --


NOTE: Your program should output the following information to a file - new_characters.txt.

Wonder Woman
Diana Prince
h 5 5 0 0 90
Batman
Bruce Wayne
h 7 3 0 4 62
The Joker
Jack Napier
v 5 1 0 4 80
Superman
Clark Kent
h 8 4 1 3 0
Catwoman
Selina Kyle
v 12 0 6 6 50
Aquaman
Arthur Curry
h 8 2 2 4 30
Iron Man
Tony Stark
h 12 8 4 0 23
Hulk
Bruce Banner
h 9 4 1 4 55
Thanos
n/a
v 10 2 0 8 90
```

USEFUL BUILT-IN PYTHON FUNCTIONS – REQUIRED FOR PART II

Formatting Integers (useful for part II):
You can use the `format()` function to format the way integers and strings are displayed to the screen.
In the following example:

```
print(format(total_user_score, '10d'))
```

the integer value assigned to variable `total_user_score` is printed in a field that is 10 spaces wide. By default, it is right-aligned within the field width.

There are other alignment options as follows:

Option	Meaning
`'<'`	Forces the field to be left-aligned within the available space (this is the default for most objects).
`'>'`	Forces the field to be right-aligned within the available space (this is the default for numbers).
`'^'`	Forces the field to be centered within the available space.

More examples of use (including output):
```
>>> total_user_score = 100
>>> format(total_user_score, '10d')
'       100'
>>> format(total_user_score, '^10d')
'   100    '
>>> format(total_user_score, '<10d')
'100       '
>>> format(total_user_score, '>10d')
'       100'
```

Use nested with the print function:
```
>>> total_user_score = 100
>>> print(format(total_user_score, '10d'))
       100
>>> print(format(total_user_score, '^10d'))
   100
>>> print(format(total_user_score, '<10d'))
100
>>> print(format(total_user_score, '>10d'))
       100
```

Formatting Text (aligning the text and specifying a width)
Examples of use, nested with the print function (including output):

```
>>> format("You", '10s')
'You       '
>>> format("You", '^10s')
'   You    '
>>> format("You", '<10s')
'You       '
>>> format("You", '>10s')
'       You'

>>> print(format("You", '10s'))
You
>>> print(format("You", '^10s'))
   You
>>> print(format("You", '<10s'))
You
>>> print(format("You", '>10s'))
       You
```