CSE101

DUE MAY 10

23:59 KST

EXTRA CREDIT I

PROFESSOR FRANCOIS RAMEAU

SEARCH & SORT

General instructions

- Please add your name and email in the dedicated location at the top of your code
- Do not use any external library, except when explicitly requested
- Try to follow the naming convention proposed by PEP-8 seen in class
- Use ONLY what we have already seen in class. If not, you will get ZERO points
- Use meaningful names for your variables and functions
- If you face problem submitting your code via GitHub please contact the professor and the TA by email
- Note that the received code will be tested on a classifier to detect potential usage of Large Language Model. We will also pay particular attention to plagiarism
- Leave comments in your code to explain your code and describe the difficulties you faced

INVITATION LINK

https://classroom.github.com/a/shfyyksJ

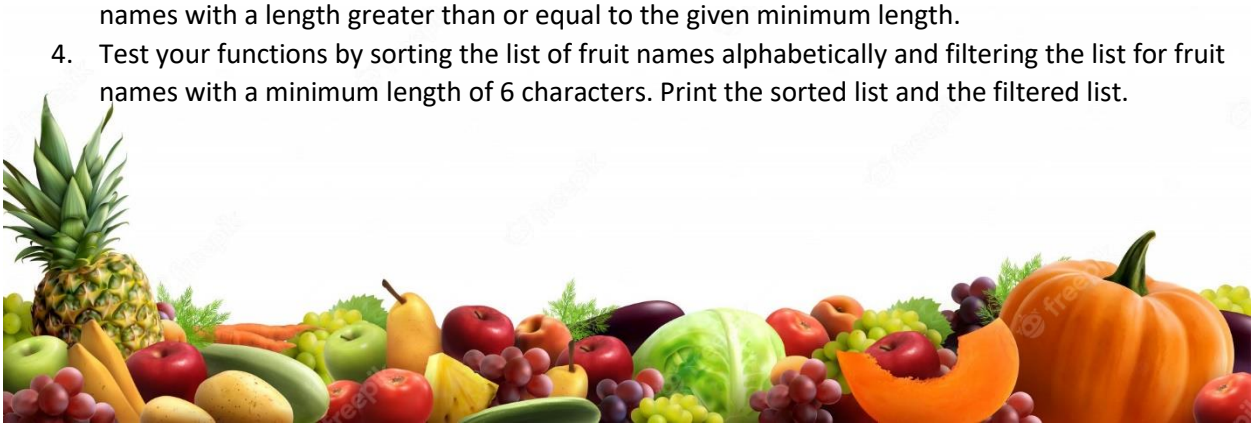
Exercise 1: Sorting and Filtering a List of Fruit (4 points)

Objective:

In this exercise, you will practice sorting a list of fruit names alphabetically and filtering the list based on a given length.

Instructions:

1. Create a list of 10 different fruit names. For example: `fruits = ['apple', 'banana', 'orange', 'kiwi', 'strawberry', 'grape', 'mango', 'blueberry', 'pineapple', 'watermelon']`
2. Write a function `sort_fruits_alphabetically(fruits)` that takes the list of fruit names as input and sorts it alphabetically using the built-in `sorted()` function. Return the sorted list.
3. Write a function `filter_fruits_by_length(fruits, min_length)` that takes the list of fruit names and a minimum length as input, and returns a new list containing only the fruit names with a length greater than or equal to the given minimum length.
4. Test your functions by sorting the list of fruit names alphabetically and filtering the list for fruit names with a minimum length of 6 characters. Print the sorted list and the filtered list.

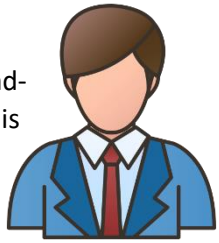


Recap

1.	Declare a list of fruits	1 point
2.	Sort fruit using built-in function	1 point
3.	Filter fruits based on string length	1 point
4.	Final test & print	1 point

Exercise 2: Second Hand Bookstore Search and Sorting (6 points)

Background story:



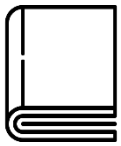
You are an avid entrepreneur ready to compete with Amazon in the market of second-hand books. To promote your company, you have plenty of brilliant ideas. Your goal is to create a user-friendly platform that will gain popularity. To make this a reality, your first step is to develop a Python program that effectively searches and sorts books from a large dataset.

To implement this code, you have collected a representative dataset of books that you have stored in a CSV file. Each element in this list is a book that is characterized by the following attributes:

Title
String

Price
Float

Quality
Integer



Hardcover
Bool

As one of the first steps of your development, you would like to be able to perform the following actions:

1. Read your database
2. Return all the books given a title input
3. Sort these books per price
4. Return all the books in a given quality range

1. Read the database (0 point)

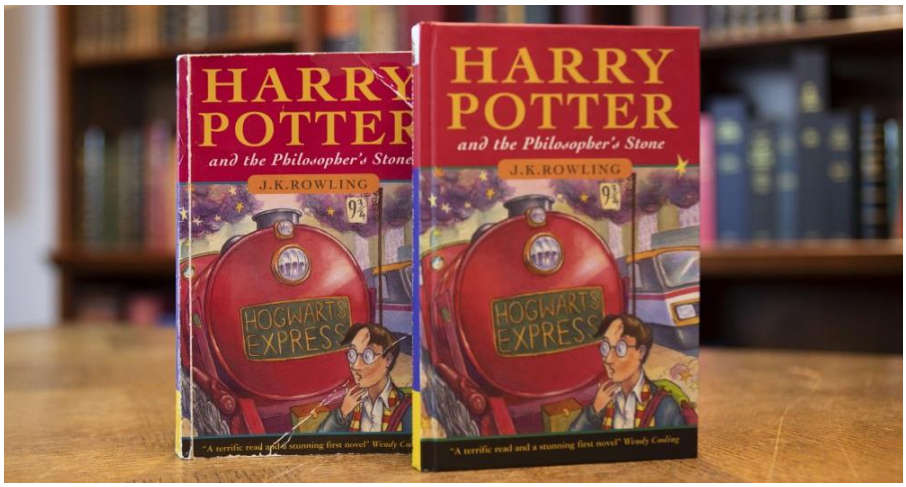
In this assignment, you are provided with a representative dataset containing 72 books which the features described before. A function designed to open this CSV file (books.csv) is also directly available to you. Each row in the file represents a book with the following information: Title (string), Price (float), Quality (int, ranging from 0 to 5), and Soft Cover (boolean).

TODO: Call the function `read_books_from_csv(filename)` to read the database with the proper path to open your file.

2. Return all the books with a given title (1 point)

In your second-hand book dataset, you can have multiple occurrences of the same book with different prices, quality, etc. Your first step will be to return to the list of books having the title entered by the user. For instance, if the user is interested in "Harry Potter", you should return a list containing all the 27 occurrences of the Harry Potter book.

TODO: Write a function `searchBookTitle(list_book, title)` that takes a list of books and a title as input and returns a new list containing all the books with the given title.

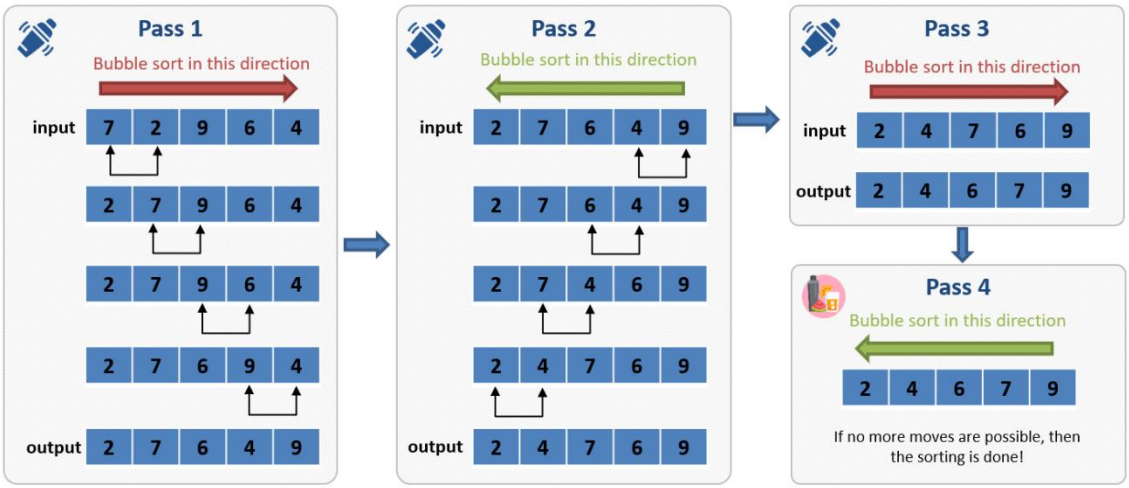


3. Sort the book by price or quality (4 point)

Sorting is of primary importance on e-commerce platforms. Therefore you would like to develop a very versatile function were the user can sort either by price or quality and in ascending or descending order.

In this exercise, you will have to implement the *cocktail shaker sort*. It is a rather simple sorting algorithm but is relatively ineffective as it admits an n^2 complexity, but it will be enough for your prototype.

Cocktail shaker sort, also known as bidirectional bubble sort, is a variation of the bubble sort algorithm that sorts a list by comparing and swapping adjacent elements in alternating passes through the list, first left to right and then right to left. This two-directional approach allows for both the smallest and largest unsorted elements to be efficiently bubbled towards their respective positions in each iteration, thus potentially speeding up the sorting process.



You should create a function `shakerSortBooks(list_book, what_to_sort, ascending)` that takes a list of books, a string indicating whether to sort by "price" or "quality", and a boolean indicating whether the sorting should be in ascending or descending order. Please refer to the comments in the code for details regarding these inputs. In the main, you will use this function to sort the “Harry Potter” books by price in ascending order.

It is the most difficult question, if you face too much difficulties, replace shaker sort by bubble sort.

4. Return books in a range provided by the user (1 point)

Write a function `rangeQualityCheckBooks(list_book, quality_range)` that takes a list of books and a range (given as a list with two integers) as input and returns a new list containing only the books with a quality score within the specified range (exclusive).

Recap

1.	Read the database	0 point
2.	Search books per title	1 point
3.	Sort books by price	4 point
4.	Return sorted Harry Potter books in a quality range	1 point