# Python Challenge

1. Implement a program that reads a text file and counts the occurrences of each word, ignoring case sensitivity.

2. Write a Python function that takes a list of strings as input and returns a new list with the strings sorted in descending order of their lengths.

3. Write a function that takes a list of numbers as input and returns the second-largest number.

4. Write a Python program that takes a list of integers as input and returns a new list with only the numbers that are prime.

5. Write a Python function that takes a list of integers as input and returns a new list with only the numbers that are perfect squares.

6. Write a Python function that takes a list of numbers as input and returns the sum of all the numbers divisible by 3 or 5.

7. Write a Python function called calculate_discounted_price that takes the original price of an item and a discount percentage as input. The function should return the discounted price after applying the discount. Ensure that the function handles the case where the discount percentage is negative and raises a custom exception called InvalidDiscountError with an appropriate error message.

8. Write a function that takes a sentence as input and returns a new sentence with the words reversed, while keeping the order of the words in the sentence.

9. Implement a program that simulates a basic calculator, allowing users to perform arithmetic operations (addition, subtraction, multiplication, division) on two numbers.

10. Create a class named Notes for handling text-based file operations.
    Class should contain methods "write", "read" and then "append" as instance methods or class methods. (Can contain any other methods if you wish)

    Use a single file for saving the notes. You can set the file name as a constant somewhere in the program (Or as a class variable).

    **write** method should create the if it doesn't exist, Then it should overwrite the older contents with the user input if the user plans to overwrite the file.
    **read** method should read the whole file contents and return it. If the file doesn't exist, then it should return "No notes found"
    **append** method should take the user input value and it must add the value to the end of the file. It must not overwrite the file.

    Now create a program to utilize this class. The program should repeatedly ask the user for these 4 choices :
    1 - Write Note (Overwrite existing).
    2 - Add more Notes (Append).
    3 - Read Notes.
    4 - Exit.