Warm-up: 1

Write a function that takes a string as input and returns the number of vowels in the string.

Warm-up: 2

Write a function that takes two lists as input and returns the common elements in both lists.

Warm-up: 3

Write a function that takes a list of numbers as input and returns the sum of the numbers in the list.

Warm-up: 4

Write a function that takes a list of strings as input and returns the longest string in the list.

Warm-up: 5

Write a function that takes both positional and keyword arguments

to welcome Smith and Abraham

Warm-up: 6

Write a Python function that takes an arbitrary number of numbers as arguments and returns the average of the numbers.

Warm-up: 7

Write a function that takes two numbers as input and returns the greatest common divisor (GCD) of the two numbers.

Warm-up: 8

Write a function that takes a string as input and returns the reverse of the string.

Warm-up: 9

Write a function that takes a string as input and returns a new string with all the spaces removed.

Warm-up: 10

Write a function that takes a list of numbers as input and returns a new list with all the duplicates removed.

Exercise 1

Write a function **calculateCurve** that takes a score as an argument and returns the total score after adding a curve. The curve is calculated based on the score as follows:

- If the score is less than 0 or greater than 100, raise a **ValueError** with the message "Score must be between 0 and 100".
- If the score is equal to or greater than 90, the curve is 5.
- If the score is equal to or greater than 80 but less than 90, the curve is 10.
- If the score is less than 80, the curve is 20.

Write a **try-except** block to call the **calculateCurve** function with the argument **50** and print the total score. If a **ValueError** is raised, print the error message.

Exercise 2

- Write a program that contains two functions: **isMultipleOfFive** and **multipleOfFive**.

- The **isMultipleOfFive** function takes an integer as an argument and returns **True** if it is a multiple of 5, and **False** otherwise.

- The **multipleOfFive** function takes no arguments. It repeatedly prompts the user to enter an integer until a valid integer is entered.

- Then it calls the **isMultipleOfFive** function to check if the entered integer is a multiple of 5. If it is, the function prints the message **[integer] is a multiple of 5**. If not, the function prints the message **[integer] is not a multiple of 5**.

- Finally, call the **multipleOfFive** function to test it.

Exercise 3

- Write a program that contains a function **calculateHypo**.

- The function takes no arguments. It repeatedly prompts the user to enter two valid float values for the two sides of a right triangle.

- The function uses the **math.hypot** function from the **math** module to calculate the length of the hypotenuse of the triangle using the two sides.

- The function returns the length of the hypotenuse rounded to two decimal places.

- Finally, call the **calculateHypo** function and print the length of the hypotenuse.

Exercise 4

- Write a program that contains a function **play**. The function takes no arguments. It repeatedly prompts the user to enter a positive integer that represents the number of times they want to play a coin tossing game.

- The function uses the **random.randint** function from the **random** module to simulate a coin toss for each play. If the result of a coin toss is 1, player 1 earns 10 points. If the result is 2, player 2 earns 10 points.

- The function keeps track of the total points for each player and displays the total points for both players at the end of the game.

- The function also displays the winner or if the game is a tie.

- Finally, call the **play** function to play the game.

Exercise 5

- Write a function **factorial** that takes a non-negative integer as an argument and returns its factorial. If the argument is negative, the function returns the string "Enter a non-negative value!".

- The factorial of a non-negative integer **n** is the product of all positive integers less than or equal to **n**. For example, the factorial of 5 is 5! = 5 x 4 x 3 x 2 x 1 = 120.

- Write a program that calls the **factorial** function with the argument **10** and prints the factorial. If the result is not an integer, print the string returned by the function.

Exercise 6

Write a program that contains two functions: **check_grade** and **get_score**.
The **check_grade** function takes a score as an argument and returns the corresponding letter grade based on the following scale:

- A for scores 90 and above
- B for scores 80 and above
- C for scores 70 and above
- D for scores 60 and above
- F for scores less than 60

The **get_score** function takes no arguments. It repeatedly prompts the user to enter a score between 0 and 100 until a valid score is entered. Then it calls the **check_grade** function to determine the letter grade for the score. Finally, it prints the letter grade and asks the user if they want to continue.

If the user enters 'y', the process repeats. If the user enters 'n' or any other value, the program terminates.

In case of an invalid input (e.g., a non-numeric value), the program should catch the **ValueError** and print an error message "Invalid input. Only numbers are allowed.". Then the program should call the **get_score** function again to continue the process.

Finally, call the **get_score** function to test it.