Additional Exercise questions:

Question no-1

a) Write a program that converts a binary (base 2) number to decimal (base 10). Your program should begin by reading the binary number from the user as a string. Then it should compute the equivalent decimal number by processing each digit in the binary number. Finally, your program should display the equivalent decimal number with an appropriate message.

For Example:

1. First, write the given binary number and count the powers of 2 from right to left (powers starting from 0)
2. Now, write each binary digit (right to left) with the corresponding powers of 2 from (right to left), such that first binary digit (MSB) will be multiplied with the greatest power of 2.
3. Add all the products in the above step
4. The final answer will be the required decimal number

Given binary number = (1101)
Now, multiplying each digit from MSB to LSB reduces the power of the base number 2.
$1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$
$= 8 + 4 + 0 + 1$
$= 13$

Test Case 1: Input: 10011010010 Output:1234	**Test Case 2:** Input:11111111 Output: 255	**Test Case 3:** Input: 1111 Output:15

Question no-2

Write a function that takes a string of characters as its first parameter, and the width of the terminal in characters as its second parameter. Your function should return a new string that consists of the original string and the correct number of leading "-"so that the original string will appear centered within the provided width when it is printed. Do not add any characters to the end of the string. Include a main program that demonstrates your function. Note: width must be greater than equal to length of string.

Test Case 1: Input: func("Hello",10) Output: ---Hello--	**Test Case 2:** Input: func("Hello",19) Output: -------Hello-------	**Test Case 3:** Input: func("Hello",2) Output: Invalid width

Question no-3

Suppose you are given a list of dictionaries representing the details of a group of employees in a company. Each dictionary has the following keys:
name: a string representing the name of the employee.
age: an integer representing the age of the employee.
salary: a float representing the salary of the employee.
Write a Python program that takes this list of dictionaries as input and performs the following tasks:

i) Use filter() and a lambda function to find all employees who are above the age of 30.
ii) Use map() and a lambda function to increase the salary of all employees by 10%.
iii) Use filter() and a lambda function to find all employees whose salary is above 50000.

The program should then print the resulting lists for each of the three tasks.

Test Case 1:
Input:

```
employees = [
    {"name": "John Doe", "age": 25, "salary": 40000.0},
    {"name": "Jane Doe", "age": 35, "salary": 60000.0},
    {"name": "Bob Smith", "age": 45, "salary": 80000.0},
    {"name": "Alice Johnson", "age": 30, "salary": 55000.0},
    {"name": "Mike Williams", "age": 28, "salary": 45000.0},
]
```

Output:
Employees above the age of 30:

```
[{'name': 'Jane Doe', 'age': 35, 'salary': 60000.0}
{'name': 'Bob Smith', 'age': 45, 'salary': 80000.0}
{'name': 'Alice Johnson', 'age': 30, 'salary': 55000.0}]
```

Employees with increased salary:

```
[{'name': 'John Doe', 'age': 25, 'salary': 44000.0}
{'name': 'Jane Doe', 'age': 35, 'salary': 66000.0}
{'name': 'Bob Smith', 'age': 45, 'salary': 88000.0}
{'name': 'Alice Johnson', 'age': 30, 'salary': 60500.0}
{'name': 'Mike Williams', 'age': 28, 'salary': 49500.0}]
```

Employees with salary above 50000:

```
[{'name': 'Jane Doe', 'age': 35, 'salary': 60000.0}
{'name': 'Bob Smith', 'age': 45, 'salary': 80000.0}
{'name': 'Alice Johnson', 'age': 30, 'salary': 55000.0}]
```

Test Case 2:
Input:
employees = [
{"name": "John Smith", "age": 20, "salary": 35000.0},
{"name": "Emily Brown", "age": 32, "salary": 45000.0},
]
Output:
Employees above the age of 30:
[{'name': 'Emily Brown', 'age': 32, 'salary': 45000.0}]
Employees with increased salary:
[{'name': 'John Smith', 'age': 20, 'salary': 38500.0}
{'name': 'Emily Brown', 'age': 32, 'salary': 49500.0}]
Employees with salary above 50000:
[]

Question no-4

A particular zoo determines the price of admission based on the age of the guest.
Guests 2 years of age and less are admitted without charge.
Children between 2 and 12 years of age cost $14.00. [Age 12 inclusive]
Seniors aged 65 years and over cost $18.00.
Admission for all other guests is $23.00.
Create a program that begins by reading the number of guests and the ages of all of the guests in a group from the user. Then your program should display the admission cost for the group with an appropriate message. The cost should be displayed using two decimal places.

Test Case 1:	**Test Case 2:**	**Test Case 3:**
Number of guests : 4 Visitor 1 age : 4 Visitor 2 age : 1 Visitor 3 age : 68 Visitor 4 age : 34 Output:$55.00.	Number of guests : 5 Visitor 1 age : 22 Visitor 2 age : 1 Visitor 3 age : 65 Visitor 4 age : 33 Visitor 5 age : 2 Output:$64.00.	Number of guests : 3 Visitor 1 age : 10 Visitor 2 age : 12 Visitor 3 age : 34 Output:$51.00.

Question no-5

Write a Python program that takes a list of tuples ***test_list*** and divisor ***K*** as input. The program should then find and print all tuples in the list where all elements in the tuple are divisible by ***K***.

Test Case 1: Input: test_list = [(6,24),(60,12), (18,21)], K=6 Output: [(6,24),(60,12)]	**Test Case 2:** Input: test_list = [(60,20),(15,10), (18,21)], K=5 Output: [(60,20),(15,10)]	**Test Case 3:** Input: test_list = [(14,21),(49,63), (18,21)], K=7 Output: [(14,21),(49,63)]

Question no-6

Write a Python function named ***check_divide*** that takes a number as an input and checks whether the digits of the number divide it and return ***Yes*** if it divides else ***No***.
Example: 128%1 = 0, 128%2 = 0, 128%8 = 0

Test Case 1: Input: check_divide(128) Output: Yes	**Test Case 2:** Input: check_divide(130) Output: No	**Test Case 3:** Input: check_divide(480) Output: Yes

Question no-7

a) Write a Python program to count the positive, negative and zero numbers from a list.

Test Case 1: Given x = [23, 4, -6, 23, -9, 21,0, 3, -45, 0,-8] Result: Positive: 5 Negative: 4 Zero: 2	**Test Case 2:** Given x = [23, 4, 23, 21,0, 3, 0] Result: Positive: 5 Negative: 0 Zero: 2	**Test Case 3:** Given x = [23, 4, -6, 23, -9, 21, 3, -45, -8] Result: Positive: 5 Negative: 4 Zero: 0

Question no-8

b) Write a Python program to find all three and/or four character words which end with vowels in a string.

Test Case 1: Input: "The quick brown fox jumps over the laze dog in an ice." Output: ['The', 'the', 'laze' ,'ice']	**Test Case 2:** Input: "The little bird flew across the blue sky and landed on the soft green grass." Output: ['The', 'blue', 'the']

Question no-9

a) Write a Python program that takes two tuples ***test_tuple1*** and ***test_tuple2*** and returns a list of all pair combinations of the two tuples.

Test Case 1: Input: test_tuple1(7,2) test_tuple2(7,8) Output : [(7,7),(7,8),(2,7),(2,8),(7, 7),(7,2),(8,2)]	**Test Case 2:** Input: test_tuple1(4,5,6) test_tuple2(7,8) Output : [(4, 7), (4, 8), (5, 7), (5, 8), (6, 7), (6, 8), (7, 4), (7, 5), (7, 6), (8, 4), (8, 5), (8, 6)]	**Test Case 3:** Input: test_tuple1(4,5,6) test_tuple2(2,3,4) Output : [(4, 2), (4, 3), (4, 4), (5, 2), (5, 3), (5, 4), (6, 2), (6, 3), (6, 4), (2, 4), (2, 5), (2, 6), (3, 4), (3, 5), (3, 6), (4, 4), (4, 5), (4, 6)]

Question no-10

b) Write a Python program that takes two lists, ***lst_1*** and ***lst_2*** as input and performs the following using lambda and filter functions:

i) Intersection of the two lists
ii) lst_1 - lst_2 (Elements present in lst_1 but not in lst_2 without any duplicates)
iii) lst_2 - lst_1 (Elements present in lst_2 but not in lst_1 without any duplicates)
iv) Union of the two lists

Test Case 1: lst_1 = [1,6,7,10,13,32] lst_2 = [13,17,18,32] **Output:** i) [13,32] ii) [1,6,7,10] iii) [17,18] iv) [1,6,7,10,13,32,17,18]	**Test Case 2:** lst_1 = [1,6,7,10] lst_2 = [13,17,18,32] **Output:** i) [ ] ii) [1,6,7,10] iii) [13,17,18,32] iv) [1,6,7,10,13,17,18,32]

Question no-11

a) Write a program to swap every odd place and even place numbers from the tuple of 10 numbers. Display user the numbers before swapping and after swapping.

Test case 1: **Input:** (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) **Output:** **Original tuple:** (10, 20, 30, 40, 50, 60, 70, 80, 90, 100) **Tuple after swapping:** (20, 10, 40, 30, 60, 50, 80, 70, 100, 90)	**Test case 2:** **Input:** (1, 3, 5, 7, 9, 11, 13, 15, 17, 19) **Output:** **Original tuple:** (1, 3, 5, 7, 9, 11, 13, 15, 17, 19) **Tuple after swapping:** (3, 1, 7, 5, 11, 9, 15, 13, 19, 17)

Question no-12

Write a program that takes a list of tuples called *test_list*. The output is a list of tuples containing one tuple which is joined with a similar initial element, and the remaining tuples.

Test Case 1: Input: test_list = [(5, 6), (5, 7), (5, 8), (6, 10), (7, 13)] Output : [(5, 6, 7, 8), (6, 10), (7, 13)]	**Test Case 2:** Input: test_list = [(5, 6), (6, 7), (6, 8), (6, 10), (7, 13)] Output : [(5, 6), (6, 7, 8, 10), (7, 13)]	**Test Case 3:** Input: test_list = [(1,3),(4,5),(6,7)] Output: [(1,3),(4,5),(6,7)]

Question no-13

b) Write a Python function named "find_average" that takes in a list of integers as a parameter and returns the average of those numbers. The function takes in an additional boolean parameter named "rounded". If this parameter is set to True, the function should return the average rounded to the nearest whole number. If the parameter is False or not provided, the function should return the average as a float with two decimal places.

Test Case 1: Input: find_average(2,3,4,5,rounded=True) Output : 4.0	**Test Case 2:** Input: find_average(2,3,4,5) Output : 3.5	**Test Case 3:** Input: find_average(2,3,4,5,6,7, rounded=False) Output : 4.5

Question no-14

a) Using the function Take input of n-digit binary number (N) through Keyboard. Find out the sum multiplication of each of the digits of N by increasing the power (starting from 0) of n from left to right.

Ex: N = 10110 (5 digit) (Sum = 1 * 5^0 + 0 * 5^1 + 1 * 5^2 + 1 * 5^3 + 0 * 5^4 = 151)
N=001 (3 digit) (Sum = 0 * 3^0 + 0 * 3^1 + 1 * 3^2 = 9)

Test Case 1: Input: 101 Output: 10	**Test Case 2:** Input: 10110 Output: 151	**Test Case 3:** Input: 0 Output: 0

Question no-15

Write a recursive function in Python that takes an integer as input and returns the n^{th} number in the Fibonacci sequence. The Fibonacci sequence is defined as follows: the first two numbers in the sequence are 0 and 1, and each subsequent number is the sum of the previous two. For example, the first ten numbers in the Fibonacci sequence are 0, 1, 1, 2, 3, 5, 8, 13, 21, and 34.

Test Case 1: Input: fibo(7) Output : 8	**Test Case 2:** Input: fibo(100) Output : 354224848179261915075	**Test Case 3:** Input: fibo(0) Output : 0	**Test Case 4:** Input: fibo(-1) Output : "Invalid Input"

Question no-16

a) Write a program in python that takes a list of tuples *test_list* and a range *(i,j)* as input. It filters records in such a way that records that do not contain an exact number of elements and lie in a range *(i,j)* are discarded. Use *filter* and *lambda* functions.

Test Case 1: Input: test_list = [(4,), (5, 6), (2, 3, 5), (5, 6, 8, 2), (5, 9)] i=2,j=3 Output : [(5, 6), (2, 3, 5), (5, 9)]	**Test Case 2:** Input: test_list = [(4,), (5, 6), (2, 3, 5), (5, 6, 8, 2), (5, 9)] i=1,j=2 Output : [(4,),(5,6),(5,9)]	**Test Case 3:** Input: test_list = [(4,), (5, 6), (2, 3, 5), (5, 6, 8, 2), (5, 9)] i=5,j=6 Output : [ ]