

Exercise 6: Create a recursive function 12 min

Write a program to create a **recursive function to calculate the sum of numbers** from 0 to 10.

- **Hint:** A recursive function is a function that calls itself again and again.

Expected Output:

55

Exercise 7: Write a Python function to check whether a string is a pangram or not. 10 min

Hint : Pangrams are words or sentences containing every letter of the alphabet at least **once**.

Given:

```
The quick brown fox jumps over the lazy dog
```

Expected output:

```
True
```

Exercise 8: Write a Python function to check whether a number is perfect or not. 10 min

- **Hint:** search about Perfect number

Given:

```
perfect_number(6)
```

Expected output:

```
True
```

Exercise 9: Write a Python program to check if a number is a power of a given base. 10 min

```
print(isPower(127,2))
False
print(isPower(128,2))
True
print(isPower(27,2))
False
print(isPower (27,3))
True
print(isPower (28,3))
False
print(isPower (2**10,2))
True
print(isPower (2**12,2))
True
print(isPower(2,2))
True
print(isPower(5,5))
True
print(isPower(10,1))
False
```

Exercise 10: Create an inner function to calculate the addition in the following way. 12 min

- Create an outer function that will accept two parameters, `a` and `b`
- Create an inner function inside an outer function that will calculate the addition of `a` and `b`
- At last, an outer function will add 5 into addition and return it

Hint:

```
def outer_fun(a, b):  
    pass    You, 1 second ago • Uncommitted changes  
    # inner function  
    def addition(a, b):  
        pass
```

For example:

```
print(outer_fun(5,10))
```

Expected output:

```
20
```

Exercise 11: Write a Python function that prints out the first n rows of Pascal's triangle. 12 min

[1]

[1, 1]

[1, 2, 1]

[1, 3, 3, 1]

[1, 4, 6, 4, 1]

[1, 5, 10, 10, 5, 1]