WEEK 2 - CONTEST

NOTE:

- **No need to submit anywhere, just keep track of all the PDF you made in a specific folder.**
- **Compare your solution with the solution I'll provide, in case of doubts, kindly reach out to me.**
- **You may get assignment solution in format of PDF or VIDEO solution, depending on the difficulty level.**

Q1. Create a function named as **checkPrime** that takes an integer as an argument. Print **YES** if the number passed is a prime number else print **NO.**

Q2. Print all the prime numbers between 1 to 100.

Q3. Create a function named **sumNum(),** which takes 2 parameters as **n1** and **n2**. Calculate and return the sum of all the numbers divisible by and 2 and 7 between **n1 to n2.** Also if the sum is **0,** then return -1.

```
# Example 1
r = sumNum(1,30)
print(r)

# Output
42

# Explanation
14 + 28

# Example 2
r = sumNum(1,10)
print(r)

# Output
-1

# Explanation
No numbers are divisible by 2 and 7 between 1 to 10
```

Q4. Make a function named **factorial(),** which takes an integer **n.** Return the factorial of that number.

```
# Example 1
r = factorial(5)
print(r)

# Output
120

# Explanation
5 * 4 * 3 * 2 * 1 is = 120

# Example 1
r = factorial(3)
print(r)

# Output
6

# Explanation
3 * 2 * 1 is = 6
```

Q5. Make a function named **sumPattern** that takes an integer **n** as an argument from the user. And then calculate the sum of the following pattern.

```
! means factorial of that number

# Example
sumPattern(5)

Means
1/1! + 1/2! + 1/3! + 1/4! + 1/5!

# Output
1.7166666666666668
```

Q6. Create a function named **sumOfSquares**, which takes a single integer as a parameter named **n.** Return the sum of squares from **1 to n.**

```
# Example 1
r = sumOfSquares(5)
print(r)

# Output
55

# Explanation
5^2 + 4^2 + 3^2 + 2^2 + 1^2
25 + 16 + 9 + 4 + 1
55
```

Q7. Create a function named as **printPrimeFactors** that takes an integer **n** as a argument and print all the prime factors of that number.

Example if **number = 20**

Then the factors of 20 are 1,2,5,10,20.

So prime factors are 2,5 (this is the output)

```
printPrimeFactors(20)
printPrimeFactors(7)
printPrimeFactors(72)

# Outputs
2 5
7
2 3
```