ASSIGNMENT 1
LOOPING (FOR / WHILE)

NOTE:

- **No need to submit anywhere, just keep track of all the Code you made in a specific folder like in VS Code**
- **Compare your solution with the solution I'll provide, in case of doubts, kindly reach out to me after the session**

Q1. Factorial of a Number: Write a program to calculate the factorial of a given number using a loop.

Example:
Enter a number = 5
Output:
120
Explanation:
Factorial of 5 is 5 x 4 x 3 x 2 x 1

Q2. Ask a number from user. Print all the factors of a that number. (Order does not matter, just print it)

Example 1:
Enter a number = 20
Output:
1 2 4 5 10 20

Example 2:
Enter a number = 100
Output:
1 2 4 5 10 20 25 50 100

Q3. Ask a number from user. Count the number of factors of that number.

Example 1:
Enter a number = 20

Output:
6

Example 2:
Enter a number = 100
Output:
9

Q4. Ask a number from user. Print the sum of all the factors of that number.

Example 1:
Enter a number = 20
Output:
42

Example 2:
Enter a number = 100
Output:
217

Q5. Ask a number from user. Print **Yes** if that number is a prime number else print **No**.

Example 1:
Enter a number = 20
Output:
No

Example 2:
Enter a number = 19
Output:
Yes

Q6. Write a program to check if a given number is a **perfect number**. A number is called perfect if it is equal to the sum of its proper divisors (divisors excluding the number itself).

Requirements:

1. You are given a positive integer **n** (Ask from user)
2. Your task is to check if **n** is a perfect number or not using a loop.

Example Scenarios to Consider:

1. **Input**: 6
   Output: True
   Explanation: The divisors of 6 are 1, 2, 3. The sum is 1 + 2 + 3 = 6, so it is a perfect number.
2. **Input**: 12
   Output: False
   Explanation: The divisors of 12 are 1, 2, 3, 4, 6. The sum is 1 + 2 + 3 + 4 + 6 = 16, which is not equal to 12.
3. **Input**: 28
   Output: True
   Explanation: The divisors of 28 are 1, 2, 4, 7, 14. The sum is 1 + 2 + 4 + 7 + 14 = 28, so it is a perfect number.
4. **Input**: 1
   Output: False
   Explanation: The number 1 has no divisors other than itself, so it is not a perfect number.

Q7. Write a program to repeatedly sum the digits of a number until only a **single-digit** number is obtained. The process involves summing the digits of the number and then repeating the process with the result until the number is reduced to a single digit.

Requirements:

1. You are given a positive integer **n** (Ask n from user)
2. You need to find the sum of the digits of n, and if the result is greater than 9, repeat the process.
3. Continue summing the digits until the result is a single-digit number.
4. Do not use lists, strings, dictionaries, or nested loops.

Example Scenarios to Consider:

1. **Input**: 9875
   Output: 2
   Explanation:
   - Step 1: 9 + 8 + 7 + 5 = 29
   - Step 2: 2 + 9 = 11
   - Step 3: 1 + 1 = 2 (single digit obtained).

2. **Input**: 12345
   Output: 6
   Explanation:
   - Step 1: 1 + 2 + 3 + 4 + 5 = 15
   - Step 2: 1 + 5 = 6 (single digit obtained).
3. **Input**: 99
   Output: 9
   Explanation:
   - Step 1: 9 + 9 = 18
   - Step 2: 1 + 8 = 9 (single digit obtained).
4. **Input**: 0
   Output: 0
   Explanation: The sum of the digits of 0 is 0, which is already a single digit.