Assignment 4 – Instructions

Study Unit 4 - Repetition Structures

Practical 4.4 – Valid Student Number (Loop through input and validation)

Write a program that asks the user for a student number and prints whether it is a valid student number or not. A student number is valid if the sum of all the digits in the student number divided by the number of digits of the student number is 0. Repeatedly ask the user if they want to enter another student number (Yes / No). Stop the programming upon "no"
Print either "**Valid**" or "**Invalid**" as output!
If the user enters 0, print "**Thank you**"

Example: Check if 31649857 is a valid student number
To verify if **31649857** is a valid student number:
Step 1:

- Sum all the digits: 3 + 1 + 6 + 4 + 9 + 8 + 5 + 7 = 43

Step 2: Check the remainder:

- 43 % 8 != 0

This means that 31649857 is not a valid student number.
You can test 51349855 to see if the programs work, as it is a valid student number. (Do not use your own)

TAKE NOTE:

- Use f-string formatting to print the output.
- Ensure that n is larger than 0
- Submit your Python script (*.py) on CodeGrade named: Practical_4_4.py

Practical 4.5 – Perfect Numbers (Nested Loops and validation)

Write a program that asks the user for an integer N and then prints all perfect numbers up to N. A perfect number is equal to the sum of its proper divisors, excluding itself. Validate that N is a positive integer.

Example: Check if 28 is a perfect number
To verify if 28 is a perfect number:
Step 1:

- Find its proper divisors: 1, 2, 4, 7, 14.

Step 2: Sum these divisors:

- 1 + 2 + 4 + 7 + 14 = 28

Since the sum of its proper divisors equals 28, it confirms that 28 is a perfect number.

TAKE NOTE:

- Use f-string formatting to print the output.
- Ensure that n is larger than 0
- Submit your Python script (*.py) on CodeGrade named: Practical_4_4.py