String contains

- Write a function that accepts two string arguments, one called needle and the other called haystack. The function checks if the haystack contains the needle (meaning, is the needle a substring of the haystack).
  - The function returns a boolean value indicating the result.
  - Limitations
    - You are not allowed to use the String.Contains() method.
  - Examples
    - Find "snakes" in "There are Snakes on this plane" -> false.
    - Find "Snakes" in "There are Snakes on this plane" -> true.
    - Find "snakes2" in "There are Snakes on this plane" -> false.

Multiply with nine

- One algorithm to multiply a number between one and nine with nine is as follows:
  - `5 * 9 == "(5-1)" + "(9-4)" == "4" + "5" == "45"`
  - Calculate the string value of the multiplicand (5) subtracted by one
  - Calculate the string value of the multiplier (9) subtracted by the value from the last step
  - Add the two values together to get the string representation of the product of the operation
- Write a program that asks an integer (ranging from 1 to 9) from the user and then multiplies it by nine with the given algorithm. Display the result to the user.

Multiply without the *-operator

- Write a function that accepts two integer values as parameters. The function returns the product (== result of multiplication operation) of the given parameters.
- Limitations

    - You are not allowed to use the *-operator.

Print until

- Write a function that has one integer parameter. The function prints the numbers from zero to the argument (including). If the parameter value is less than zero, print `Given number too small.`.
- Examples
    - `TheMethod(0) → 0`
    - `TheMethod(4) → 0, 1, 2, 3, 4`

Ask numbers until -1

- Write a program that asks integers from the user until the given number is -1. Print out the sum and the average of the numbers (-1 not included).
- Examples
    - Given only -1 → avg: 0, sum 0
    - Given the numbers 1, 2, 3, 4, -1 → avg: 2.5, sum: 10