

**Read in CHAPTER 4:**

- Repetition Structures - introduction
- The `for` loop introduction
- The `for` loop with lists
- The `for` loop with ranges
- The `for` loop - strings
- Calculating a running total; the augmented assignment operators

## REPETITION STRUCTURE

The `for` loop

### ➤ Salary increase

An employee's annual starting salary is \$47,450. Each year the employee receives 3% increase of the salary. Write a program that displays annual salaries during first five years of an employment of this employee.

```
_21Q1_W4_CW_101 LOOP.py
YEAR: 0          SALARY: $ 47,450.00
YEAR: 1          SALARY: $ 48,873.50
YEAR: 2          SALARY: $ 50,339.71
YEAR: 3          SALARY: $ 51,849.90
YEAR: 4          SALARY: $ 53,405.39
YEAR: 5          SALARY: $ 55,007.55
>>> |
```

### Running total and ranges

### ➤ Average value

Write a program that asks a user to enter three numbers and then displays their sum and an average value. In the program use a repetition structure.

Example:

```
This program displays a sum and an average of three numbers.
Number 1- enter the number: 1526
Number 2- enter the number: 258
Number 3- enter the number: -568
The sum of entered numbers is: 1216.0
The average value of entered numbers is: 405.33
>>>
```

### Practice:

Update the program such way that the user would provide the number of numbers to work with.

```
_21Q1_w4_CW_101_100p.py
How many number do you want to enter? 5
Number 1- enter the number: 256
Number 2- enter the number: 526
Number 3- enter the number: 55
Number 4- enter the number: 11
Number 5- enter the number: 55
The sum of entered numbers is: 903.0
The average value of entered numbers is: 180.60
>>> |
```

#### ➤ Number of spaces

Write a program that asks a user to enter a sentence and then displays a number of spaces in the sentence.

Example:

```
_21Q1_w4_CW_101_100p.py
Enter a sentence: We are very happy people.
There are 4 spaces in the sentence.
>>> |
```

#### ➤ Sum of first n positive integers

Write a program with a loop that asks a user to enter a positive integer n. The program displays the sum  $1+2+3+4+\dots+n$ .

Example:

```
This program displays the sum of first n positive integers.
Enter a positive integer n: 20

1+2+...+ 20 = 210
>>> |
```

➤ **First or last digit in integer**

Write a program which asks a user to provide a number of integers to work with, to enter each integer, and then displays:

- The first digit of the integer if the entered integer is positive or zero;
- The last digit of the integer if the entered integer is negative;
- The number of non-negative and the number of negative integers entered by the user.

Example:

```
How many integers do you plan to enter? 5

Enter a positive or a negative integer: 4589
The first digit of integer is 4

Enter a positive or a negative integer: 25
The first digit of integer is 2

Enter a positive or a negative integer: -1259
The last digit of integer is 9

Enter a positive or a negative integer: 12
The first digit of integer is 1

Enter a positive or a negative integer: -258
The last digit of integer is 8

You entered 3 non-negative integers.
You entered 2 negative integers.
>>> |
```

➤ **Double saving every day**

You save money such way, that first day you save 1 cent, second day 2 cents, third day 4 cents and each following day two times more than the day before.

Write a program that asks the user how many days he/she plans to keep saving money and then displays:

- The saved money for each day in dollars;
- The saved amount for the whole time period in dollars.

Example:

```
Enter the number of days of saving: 15
Saving for the day # 1      $      0.01
Saving for the day # 2      $      0.02
Saving for the day # 3      $      0.04
Saving for the day # 4      $      0.08
Saving for the day # 5      $      0.16
Saving for the day # 6      $      0.32
Saving for the day # 7      $      0.64
Saving for the day # 8      $      1.28
Saving for the day # 9      $      2.56
Saving for the day # 10     $      5.12
Saving for the day # 11     $     10.24
Saving for the day # 12     $     20.48
Saving for the day # 13     $     40.96
Saving for the day # 14     $     81.92
Saving for the day # 15     $    163.84

The total saving for 15 days is $ 327.67
>>> |
```

➤ **Multiplication table**

Write a program that prompts a user to enter an integer and then displays the multiplication table (from 1 to 10) of the entered integers.

Possible output:

```
This table displays a multiples of an entered integer.  
Enter the integer for multiplication: 27  
  
27 x 1 = 27  
27 x 2 = 54  
27 x 3 = 81  
27 x 4 = 108  
27 x 5 = 135  
27 x 6 = 162  
27 x 7 = 189  
27 x 8 = 216  
27 x 9 = 243  
27 x 10 = 270  
>>> |
```

➤ **Sum of digits**

A) Write a program that asks a user to enter a positive integer and then displays the sum of digits of the integer.

Example:

```
This program displays the sum of digits in an entered positive integer.  
Enter a positive integer: 11243  
The sum of digits is 11  
>>> |
```

B) Update the program such way that the user can also enter a negative integer.

**Example:**

```
This program displays the sum of digits in an entered integer.  
Enter a positive or a negative integer: -123124  
The sum of digits is 13  
>>> |
```

➤ **Integer transformation**

A) Write a program that asks a user to enter an integer and transforms the integers to a string following way:

- Each even digit is replaced by the symbol 'E'
- Each odd digit is replaced by the symbol 'O'
- The negative sign is replaced by the symbol '\*'

Examples:

```
This program displays a transformation of an intered integer.
```

```
Enter a positive or a negative integer: 4545666
```

```
The integer is transformed to EOEOEEEE
```

```
>>>
```

```
RESTART: C:\Users\horaki\Desktop\TCSS141_WINTER2021\WEEK #5\2  
_21Q1_W5_CW_for loop more.py
```

```
This program displays a transformation of an intered integer.
```

```
Enter a positive or a negative integer: -1233366
```

```
The integer is transformed to *OEEOOEE
```

```
>>> |
```

B) Update the previous program such way, that the user can enter the number of integers to transform at the beginning and the program displays the transformation appropriate number of times.

```
This program displays a transformation of an intered integer.
```

```
How many integers do you want to transform? 3
```

```
Enter a positive or a negative integer: 12555
```

```
The integer is transformed to OEEOO
```

```
Enter a positive or a negative integer: -12225
```

```
The integer is transformed to *OEEOO
```

```
Enter a positive or a negative integer: -44
```

```
The integer is transformed to *EE
```

➤ **Scores - min, max, average**

Write a program that asks the user to provide a number of tests during a quarter. Then the program asks the user to provide a score for each test. Scores are provided in percentage and the score cannot be lower than 0 and larger than 100.

At the end the program displays user's minimum and maximum score and the average score.

Example:

```
This program displays info about your tests.
```

```
How many tests did you take during the quarter? 5
```

```
Test # 1 - enter the score: 78.3
```

```
Test # 2 - enter the score: 69.3
```

```
Test # 3 - enter the score: 99.4
```

```
Test # 4 - enter the score: 63.5
```

```
Test # 5 - enter the score: 84.5
```

```
Your minimum score was: 63.5 %.
```

```
Your maximum score was: 99.4 %.
```

```
Your average score was: 79.00 %.
```

```
>>>
```

➤ **Multiplication table**

Write a program that displays the multiplication table from 11 to 15. See the expected output.

```
11 x 11 = 121  11 x 12 = 132  11 x 13 = 143  11 x 14 = 154  11 x 15 = 165
12 x 11 = 132  12 x 12 = 144  12 x 13 = 156  12 x 14 = 168  12 x 15 = 180
13 x 11 = 143  13 x 12 = 156  13 x 13 = 169  13 x 14 = 182  13 x 15 = 195
14 x 11 = 154  14 x 12 = 168  14 x 13 = 182  14 x 14 = 196  14 x 15 = 210
15 x 11 = 165  15 x 12 = 180  15 x 13 = 195  15 x 14 = 210  15 x 15 = 225
>>> |
```