# Assignment #2

### General Instructions:

The assignment should be performed **independently** – no collaborations are allowed.

**Late submission** will lead to a reduction in your grade – 5 points per day.

Assignment will be uploaded to **GitHub**.

In the course’s Moodle you should go to the assignment activity and upload a text file, named:

hw-<id>.txt, where <id> is your ID number. The file should contain the URL of your repository, for example: [https://github.com/israel\_israeli/*DS\_Intro\_HW\_2*](https://github.com/israel_israeli/DS_Intro_HW_2)

### Tasks

1. write a function named reverse(sentence, reverse\_word) that takes two strings as an input and returns a new string where the first occurrence of reverse\_word in the sentence has been reverted (I.e. spelled backwards).

Return type: str.

Note:

* + The reverse\_word doesn’t have to be in the sentence. In that case you should return the following string: "The word was not found".
  + You may NOT assume that "reverse\_word" can only be a string. If “reverse\_word” is not a string, then you should return "invalid input"
  + The function is case sensitive, "Like" ≠ "like".

For example:

* + reverse("I like apples and I also like bananas", "like") should return: "I ekil apples and I also like bananas".
  + reverse(I like apples and I also like bananas", "oranges") should return: "no match word found".
  + reverse(I like apples and I also like bananas", "Bananas") should return: "no match word found".
  + reverse(I like apples and I also like bananas", 3) should return: "invalid input detected".

1. **Optional** Exercise (no grading):

Write a function named compute\_equation(equation) to compute the equation represented by a string. Return type: int, float, str (errors must be str).

Note:

* The equation valid input is: 1234567890\*/-+ (\*\* is power, 5\*\*2 = 5²).
* Even though you haven't learned about data structures in python, you may use whatever data structure you want, but it can still be done only by using for/while loops.
* The equation doesn't have to contain only the symbols mentioned above. In that case you should return the following string: "invalid input detected" 5.
* You may assume that "equation" input can only be a string.
* Dividing with zero won't be tested.

For example:

* compute\_equation ("4+2\*5-2") should return: 12. (As an int, not 12.0 which is float)
* compute\_equation ("-3+-10/2\*-3-5") should return: 7 (int).
* compute\_equation ("-3\*\*3-5") should return: -32 (int).
* compute\_equation ("50\*\*2-2500\*\*0") should return: 2499 (int).
* compute\_equation ("30\*2\*\*4/8") should return: 60 (int).
* compute\_equation ("5/2\*\*4") should return: 0.3125 (float).
* compute\_equation ("5/2\*\*4+.6875") should return: 0 (int, and not 0.0 which is float). compute\_equation ("5\*2+1-6^5+0.5") should return: "invalid input detected" (str)