Research skills assignments – week 5
Testing: Input/Output

1. Read the assignment Testing: Input/Output. Start met het onderzoek en raadpleeg meerdere bronnen.
2. Write a short summary and give some examples (max. 1 A4).
3. Include a source list of at least 3 sources you consulted for the assignment.
4. Perform assignment 2 of Testing: Input/Output with your learning team.
5. Submit the summary in PDF format to Codegrade at assignment A2W8L1.
6. Add the link of the assignment in Codegrade to your dossier.

Good luck!

Summary

I/O: Input / Output. I/O is a way to provide an argument (input) and then later get a result (output). A good example of this would be a printer. With a press of a button you provide the printer with a specific PDF (data) you want to print. And then when it processed the PDF (data) it will print the letter you would like to print. Here is the PDF or 'data' the input and the letter which comes out of the printer the output. (See more on https://www.techtarget.com/whatis/definition/input-output-I-O)

With I/O testing however there is nothing physical involved, except of course your machine. Everything will happen on the machine itself. This means that instead of a PDF you provide another kind of data. An input could be something like "Number 1: 4, Number 2: 6, Operation: Addition". Then the output would be 10. However since it's I/O testing we don't say or assume it's 10 but we are going to *check* if the output we get is equal to the expected output.

In code you could follow the example Qualified gives (https://docs.qualified.io/creating-content/challenges/guides/stdio/). In this case it's for checking if the provided number is even.

```
class Solution {
    public static boolean isEven(int num) {
        return num % 2 == 0;
    }
}
```

Source list

What is I/O and why is it important?
https://www.techtarget.com/whatis/definition/input-output-I-O

Testing I/O Transforms in Apache Beam, examples of I/O testing is used in a product.
https://beam.apache.org/documentation/io/testing/

Why a client would like I/O testing and useful strategies.
https://docs.qualified.io/creating-content/challenges/guides/stdio/

Code Analysis - Week 5

1. Analyze the two given codes below without executing them. What will be the result of the programs?

```
a_tuple = ('Never', 'gonna', 'give', 'you', 'up')
counter = 0
for x in a_tuple:
    if x[0] ==  'g':
        counter = counter + 1
    else:
        counter = counter + 2
print(counter)
```

This will most likely result in `8` as it will increase the counter by `1` if the word starts with 'g', else it will increase the counter by 2.

```
def do_something(x):
    rtuple = x,
    for i in range(2,11):
        rtuple = rtuple + ((x*i),)
    return rtuple
print(do_something(6))
```

The function will use the following formula. Since it uses a range in a for loop it will start with `rtuple = 6 + ((6*2))` and end with `rtuple = 6 + ((6*11))`. So it is adding the current result to the previous result, but as a new item in the tuple. The tuple is complete when the end of the range is reached.

```
def process_strings(strings):
    processed_strings = []
    for string in strings:
        processed_string = ""
        for char in reversed(string):
            processed_string += char
        processed_strings.append(processed_string)
    return processed_strings
```

```
def main():
    names = ["Alice", "Bob", "Charlie", "Dave"]
    processed_names = process_strings(names)
    for name in processed_names:
        print(name)

main()
```

In short, we pass a list full of names into a function and then call a function we called to reverses each name. An example being that `Alice` would become `ecilA`.
When we did that to all names in the list we return the now full array with reversed names and print it in main().