TECHNISCHE HOCHSCHULE DEGGENDORF		Matrikelnummer / Student ID:	
		Platzziffer:	Punkte: Note:
Angewandte Informatik			
Kurs:	Programming 2	**Semester:**	SS 2022
Studiengang:	Artificial Intelligence	**Dauer/Anteil:**	90 Min.
Prüfer:	Prof. Dr. Florian Wahl	**Prüfungsdatum**	18.07.2022
Hilfsmittel:	everything	**Uhrzeit**	13:30
Prüfungsart:	schr.P. zuhause 90 Min.	**Anzahl d. Blätter**	9

Before you begin:

- Ensure your document contains the correct number of pages.
- Write your student ID number on every page.
- Read all the questions to get an overview of the exam.
- All solutions must be handwritten and readable.
- Everything must be submitted within time as a single PDF file.
- Write within the boxes.
- After digitalising your paper, make sure nothing was cut off.

Good luck!

Question:	1	2	3	4	5	Total
Points:	14	40	10	20	6	90
Scored:						

TECHNISCHE HOCHSCHULE DEGGENDORF		Matrikelnummer / Student ID:	
		Platzziffer:	
Kurs:	Programming 2	**Semester:**	SS 2022
Prüfer:	Prof. Dr. Florian Wahl		

Question 1 (14 Points):

Write a program that asks for a word on the command line. After receiving input it should check whether the word spells the same independent of the reading direction. Your program is expected to be case-insensitive. If the input contains multiple words, use only the first word. Your function should return **`True`** if the word spells the same backwards, as forwards and **`False`** otherwise.

Examples:

- `Anna` will return **`True`**
- `anna` will return **`True`**
- `Peter` will return **`False`**
- `hello` will return **`False`**

TECHNISCHE HOCHSCHULE DEGGENDORF		Matrikelnummer / Student ID:	
		Platzziffer:	
Kurs:	Programming 2	**Semester:**	SS 2022
Prüfer:	Prof. Dr. Florian Wahl		

Question 2 (40 Points):

Design and implement a class for modelling a traffic light. The light should have three bulbs: one red, one yellow, and one green bulb. Bulbs should be modelled as a separate class, with one variable which is the state of the bulb. Initially the state should be False but there are two class methods: *on* and *off* to change the state.

The traffic light class should provide a variable with the current state of the light. Also, there should be a class method to check the current state of the light. Valid states are

- red
- red-green (out of order)
- yellow
- green

Implement a method to set the new state of the traffic light. The method will set the new state only if the desired transition is valid. If the new state was successfully set, it should return `True` and `False` otherwise. Also this function should control the bulbs. The valid transitions are:

- From red to green
- From green to yellow
- From yellow to red
- From any state to red-green
- From red-green to red

The initial state of new traffic lights should be red-green.

Use a test driven development approach (TDD) where you write tests to cover at least the following functionality:

1. Initialization of bulbs
2. Switching of bulbs
3. Traffic lights start in the correct initial state
4. Traffic lights adhere to the correct state transitions

TECHNISCHE HOCHSCHULE DEGGENDORF		Matrikelnummer / Student ID:	
		Platzziffer:	
Kurs:	Programming 2	**Semester:**	SS 2022
Prüfer:	Prof. Dr. Florian Wahl		

Question 3 (10 Points):

Consider you are built a poker game with a variety of functions. Now you want to apply the Model-View-Controller architecture to your code. Please mark which of the following functions you would map to which part of the architecture and explain your decisions.

#	**Function Description**	**Model**	**View**	**Controller**
1	Handle mouse clicks			
2	Compute next move			
3	Compute the next card on the dealers deck			
4	Display your current hand			
5	Check who won?			

Explanation 1:

Explanation 2:

Explanation 3:

Explanation 4:

Explanation 5:

Question 4 (20 Points):

Write a function to check the parenthesis in a source code file. The function should take the filename of a python file, open it and parse it. If all parentheses (all means (), [], and {}) are opened and closed in the correct order the function should return **`True`**. If an error is found, the line number containing the first error should be printed and **`False`** should be returned.

Examples:

```
1 i = 23
2 print((i)
```

should return **`False`** and print "Error in line 2"

```
1 i = [1]
2 print(i[0])
```

should return **`True`** and print nothing.

TECHNISCHE HOCHSCHULE DEGGENDORF		Matrikelnummer / Student ID:	
		Platzziffer:	
Kurs:	Programming 2	**Semester:**	SS 2022
Prüfer:	Prof. Dr. Florian Wahl		

Question 5 (6 Points):

Describe your three favorite suggestions from PEP-8, how they impact your future life as a developer and why you find them useful.