Requirements for video presentation (max. 10 min.)

Please note that the video presentation meets the following:

• **The video should not be longer than 10 min.**

Dia 1:

Hello, and welcome to the video presentation for the group assignment in the course computational thinking, from group 47.

Dia 2:

The problem we had to solve was to create a program and algorithm that can detect whether a provided credit card number is valid with the rules of Luhn's algorithm. Here we see the rules for the basic version.

Dia 3:

We are allowed to make our own assumptions of what is permitted, but we have to assume that the computer will generate numbers randomly or that a user will be entering a number of a few digits. If the provided input is not permitted, how will our algorithm handle such cases? We specify this in a second, more sophisticated version of our algorithm. On this slide we see the rules for the sophisticated version.

Dia 4:

This is the algorithm we have created together to solve the problem. This algorithm is for the basic and the sophisticated versions.

Dia 5:

Flowchart

Dia 6:

Pseudocode

Dia 7:

Python code

Dia 8:

We choose to use the solution strategy “The divide the problem into several subproblems”. Using this will help us simplify the problem and divide the bigger problem into smaller problems. When dividing the bigger problem into a sufficient amount of smaller problems, we go over to solving those little problems one by one. When completing this, we can say that we have solved our bigger problem.

The bigger problem has to be divided into subproblems to get the simulation program.

Dia 9:

In our case we found three subproblems. In this slide we see what those subproblems are.

Next to the subproblems are the steps in the algorithm we took to solve these problems.

Dia 10:

We worked on and divided our tasks in github. Mylène had experience in github, so she helped us to understand the workflow of github. Here you can see how much each person has contributed to the project over time.

Dia 11:

1. We made the pseudocode first for the basic version. Then, we wrote the python code based on the pseudocode. We then encountered the problem merging the basic and sophisticated code together. pseudocode differed a lot from the python code. This was because while writing the python code, some methods used in the pseudocode were not optimal in python code.
2. A general problem is that we didn’t know where to begin with writing the algorithm. We had a vague idea about which steps had to be taken. But that was not specific enough to make an always working recipe out of it, which is the case with an algorithm.
3. Since, for some of us this was the first time using python and github, there were technical difficulties with python and github.
4. There was a data type converting issue in the sophisticated code. What happened was that the credit card number was taken as a list of strings, while we wanted this to be integers.

Dia 12:

Ramy and Duco set up a meeting where they put the pseudocode and the python code next to each other and worked out the differences.

Dia 13:

We mailed our professor about this. He gave us the answer that we have to simply write out all the steps-in the right order of execution- that has to be done in order to simulate the

program. With this information we got together and we came to the algorithm described previously.

Dia 14:

Mylene already had a lot of experience working with github and python, so if the others had any questions about this they asked her. We also watched tutorials on how to use github and python.

Dia 15:

This problem was fixed by slicing the list as integers. The for loops in the odd and even method were first iterating over a list of strings, but then this was fixed to a list of integers.

Dia 16:
End the presentation with the summary.

https://youtu.be/ViTDjae7hmA link to the video presentation