Report

1. **State the names of the group members and the group number.**

*Group names:*

Ramy Ktllama

 Duco van den Bos

 Amandine Aabdaoui

 Mylène Brown-Coleman

*Group number:* 47

1. **What is the problem you are solving?**

In the late 1960s IBM scientist Hans Peter Luhn created an algorithm called; The Luhn algorithm or the Luhn formula, also know as the “mod 10” algorithm. The algorithm validates a variety of identification numbers, like credit card numbers, IMEI numbers etc., with the use of a simple checksum formula. The algorithm itself was not designed against hostile acts, but rather for protecting against accidental errors.

For our assignment we have to make a program based on The Luhn algorithm that can detect whether a provided number is a valid credit card number, following these set of rules;

1. Double every second digit from right to left. If doubling of a digit results in a two-digit number add up the two digits to get a single-digit number.

2. Now add all single-digit numbers from Step 1.

3. Add all digits in the odd places from right to left in the card number.

4. Sum the results from Step 2 and Step 3.

5. If the result in Step 4 is divisible by **10**, the card number is correct. Otherwise, the number is invalid. For instance, the sequence 4388576018402626 is not a valid credit card number, but the number 4388576018410707 is valid.

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.

1. **How have you tackled the problem (which algorithms, solution strategies, tools)**

We started off by making pseudocodes for the program individually, we then compared our pseudocodes to determine what the optimal solution was. We chose a pseudocode that was based on the solution strategy; “Divide the problem into several sub-problems”. We chose this strategy since the assignment wants us to follow a set of rules, which are by themselves sub problems, i.e. rule 1 wants to double every second digit from right to left. If doubling of a digit results in a two-digit number add up the two digits to get a single-digit number, we have reasoned backwards by solving these rules first, so that the bigger problem was easier to solve. In addition to that we used inclusion and exclusion boundaries for the rules. By combining these approaches, we had reduced the complexity of the problem, hence the problem was easier to solve.

For the documentation we used GitHub which provides a Web-based graphical interface with useful features like, management tools for projects. This platform allowed us to clarify tasks, track stages of work progress, add or delete files and edit each other’s work, all from one repository.

For the implementation of our program we used Python which is a general-purpose coding language. Python is really convenient since it allowed us to work from a dynamic workspace connected to our Git, where we could create files and directly push it to our individual branches on GitHub.

Concerning the flowchart, we used Lucid Chart, an online diagram software.

Last but not least, for the video presentation we used googled Drive and zoom. We created a PowerPoint in Google Drive, we presented this PowerPoint via a zoom call and recorded it, we then uploaded the recording onto YouTube.

1. **Which problems have you encountered in solving the problem**

The first problems we had encountered were some technical issues, setting up GitHub and Python, since not everyone had already worked with these platforms it was quite of a struggle to get everything up and working especially the dynamic workspace within Python. In addition to that we also had encountered several problems with the GitHub workflow.

Our first Python code did not correspond with the pseudocode, this was due to that the pseudocode was written for the sophisticated program including all of the rules mentioned in the assignment, where the Python code was implemented as a simple version of the program.

Another more general problem was that we did not know where to begin with writing the algorithm. We did have a vague idea about it, but we did not have sufficient knowledge about making algorithms.

Within the Python Program there were 2 issues, the data type converting issue and the functions for adding up the digits. This last issue was due to that the functions could not reach the certain type of data input.

* **data type converting issue**

1. **How have you solved those problems (which solution strategies/algorithms)?**

* **sliced integer list**
* **the input conversion**
* **zoom call for installing github**
* **video links youtube tutorials and github site**

1. **How did you split the tasks?**

* **we assigned issues to each other on github**
* **we created a project board on github where we’ve created issues each issues we gave a description and eventually a label and assign those to everyone**

1. **Specify who did what.**
2. **How much time have you spent on solving the problem**

**started 24th september,**