Lab 1: Advanced Python

SYSA 3204 – Dr. Sukhwant Sagar

Setting up the Python Project

This lab assignment will be completed by pairs of students. Submissions by individual students will not be accepted under normal circumstances. The intent is that the two students work together.

Create a new Python Application project. **Name the project** *Lab1_Group**N*** but replace ***N*** with your actual group number. Please have your **file name** as Lab1_firstname1_firstname2.py. Failure to follow this naming format makes it more difficult for me to download and grade your solution, so attention to this detail is appreciated.

Program Requirements:

For this lab, were going to be writing a simple credit card validator. When the program is complete, we're going to be able to determine whether a given card number is valid or not.

The algorithm we're going to use to verify card numbers is called the **Luhn algorithm, or Luhn formula**. This algorithm is used in real-life applications to test credit or debit card numbers as well as SIM card serial numbers.

The purpose of the algorithm is to identify potentially mistyped numbers, because it can determine whether or not it's possible for a given number to be the number for a valid card.

The way we're going to use the algorithm is as follows:

a) Remove the rightmost digit from the card number. This number is called the checking digit, and it will be excluded from most of our calculations.
b) Reverse the order of the remaining digits.
c) For this sequence of reversed digits, take the digits at each of the even indices (0, 2, 4, 6,
d) etc.) and double them. If any of the results are greater than 9, subtract 9 from those numbers.
e) Add together all the results and add the checking digit.
f) If the result is divisible by 10, the number is a valid card number. If it's not, the card number is not valid.

It should be able to accept a card number from the user. For this project, you can assume that the number will be entered as a single string of characters (i.e. there won't be any spaces between the numbers). However, you should be able to accept a card number with spaces at the start or end of the string.

In addition to that, your program should be able to handle following test cases:

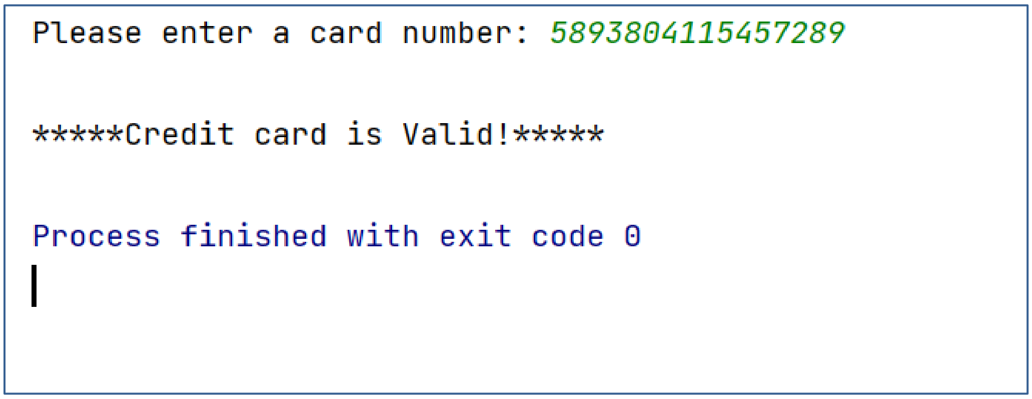
1. Valid Card number (spaces at front and end stripped off)
2. Invalid Card number
3. Empty Card number
4. Mix input Card number
5. Card number not greater than 16 digits

The program should validate that card number using the Luhn algorithm described above.

When testing your solution, you can use your own card number, or you can find valid card numbers online that are used for testing payment methods. For example, Stripe has a range of test card numbers you can use.

Expected Output:

a) **Test Case 1: Valid Credit card**

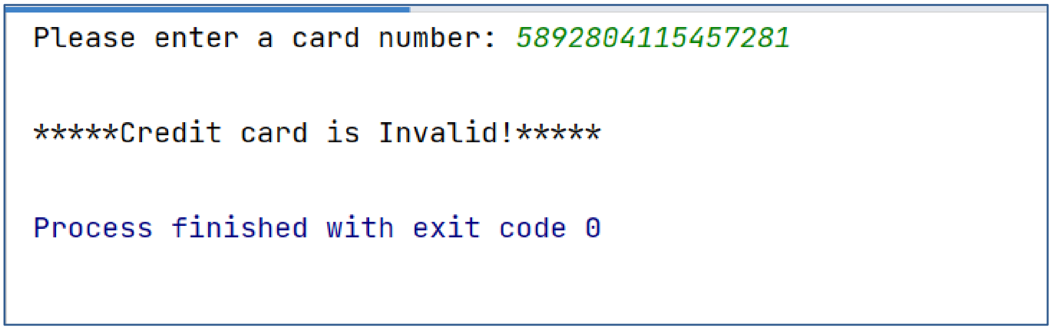


```
Please enter a card number: 5893804115457289


*****Credit card is Valid!*****


Process finished with exit code 0
```

b) Test Case 2: Invalid Credit Card

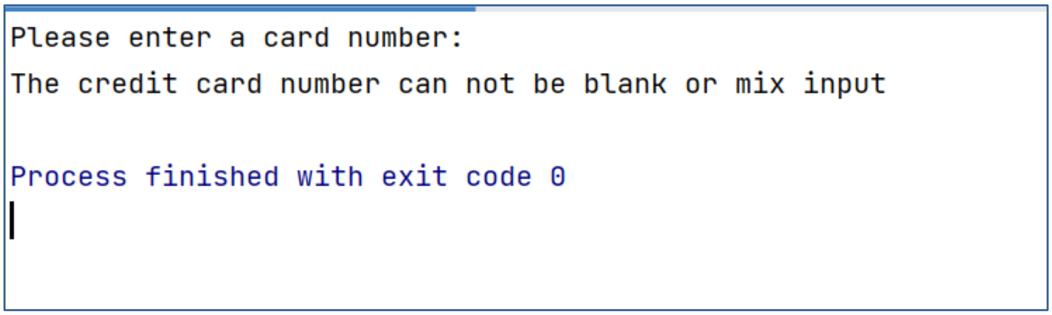


```
Please enter a card number: 5892804115457281


*****Credit card is Invalid!*****


Process finished with exit code 0
```

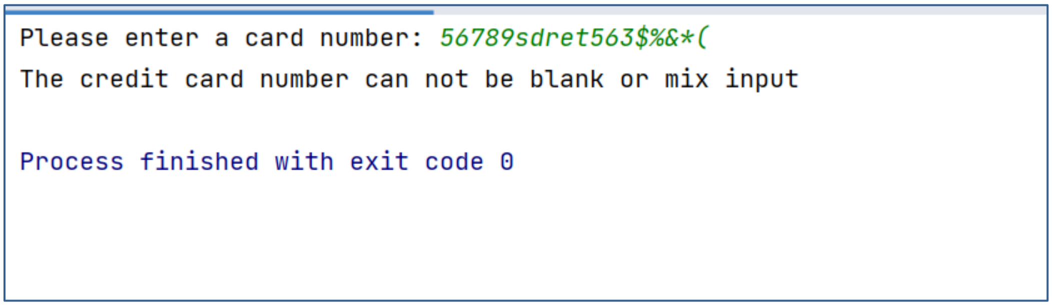
c) Test Case 3: Empty input



```
Please enter a card number:
The credit card number can not be blank or mix input

Process finished with exit code 0
```

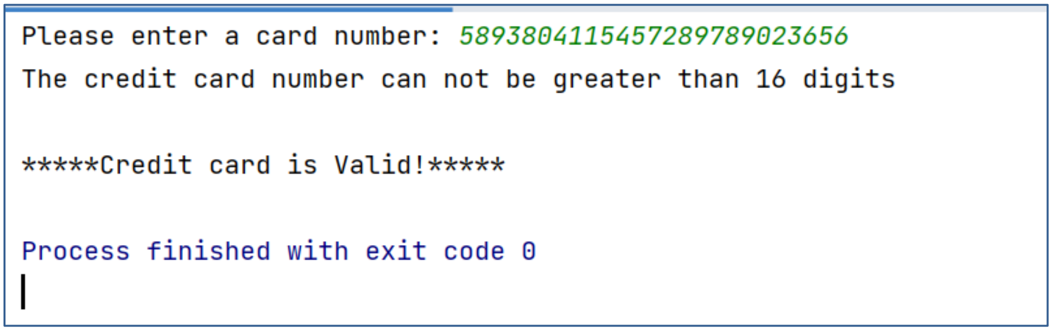
d) Test Case 4: Mix input for credit card



```
Please enter a card number: 56789sdret563$%&*(
The credit card number can not be blank or mix input

Process finished with exit code 0
```

e) **Test Case 5: Length of credit card > 16**



```
Please enter a card number: 5893804115457289789023656
The credit card number can not be greater than 16 digits

*****Credit card is Valid!*****

Process finished with exit code 0
```

Things to Explore:

You are welcome to explore beyond the mandatory requirements if you wish.

General Requirements

- Include an opening comment with your full name, the full names on the student(s) you are working with, the name of the program, the date, and a short description.
- Follow the course style guide! Use descriptive names for variables, constants, arrays, functions, etc. that follow our naming conventions.
- Attach the zipped project to the assignment folder.
- **Each of the lab partner should submit a video recording (sharing screen) explaining the code line by line along with its execution. The lab partner should also mention the distribution of tasks for this lab. Both these requirements would be considered for the teamwork component of the rubric.**