Deborah Barndt

ITMD 513 Open Source Programming

Professor Dr. Sam

Hw3

2-9-19

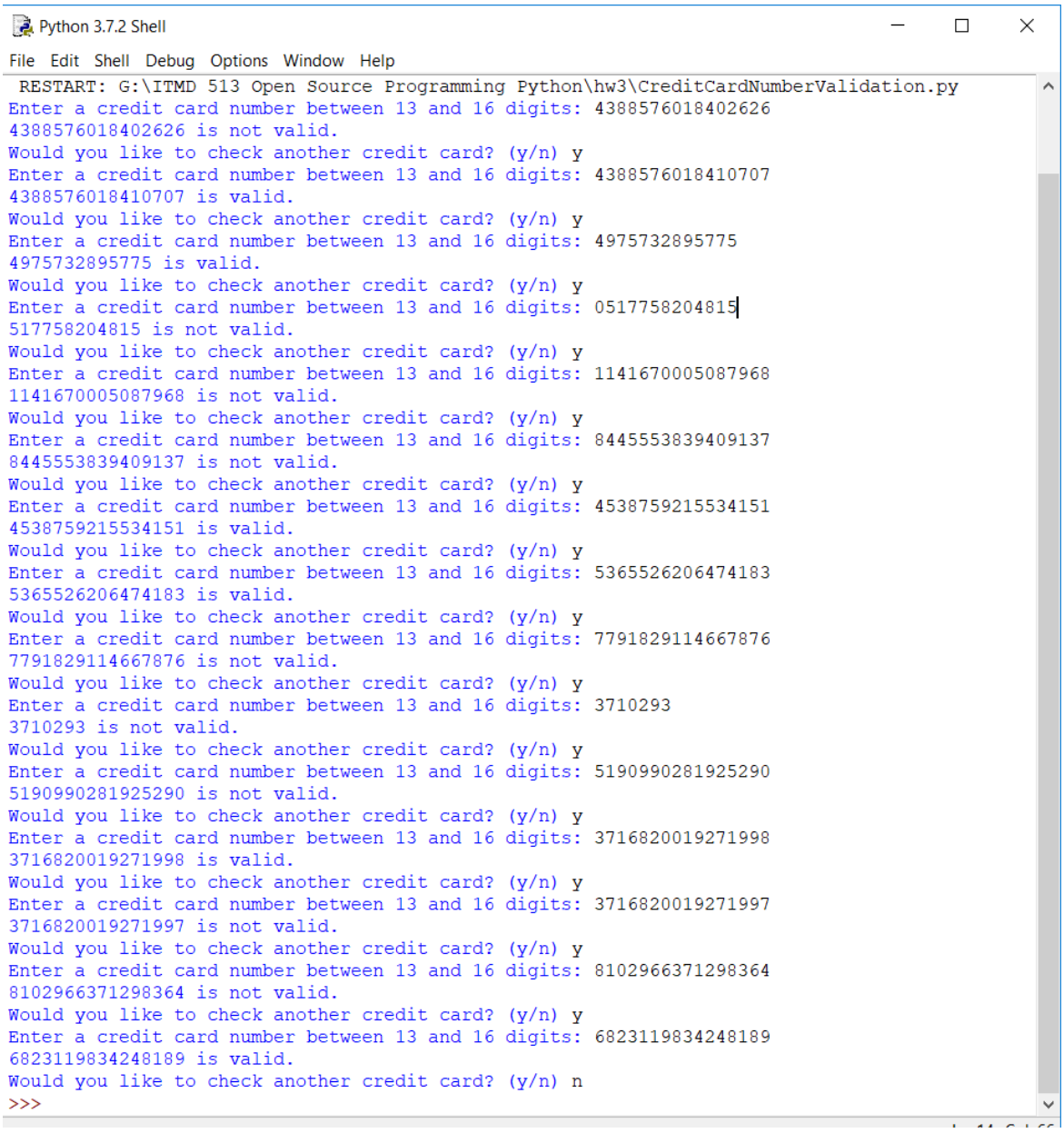
```
'''
Deborah Barndt
2-6-19
CreditCardNumberValidation.py
hw3: Financial: Credit Card Number Validation

This program will ask the user to input their credit card number, then will
use Hans Luhn's algorithm to check if the card is valid or not valid. Once the
result is displayed, it will ask the user if they would like to check another
credit card number.

The algorithm is useful to determine whether the card number is entered correctly
or whether a credit card is scanned correctly by a scanner. The following steps are
used to check the credit card numbers: 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, add all the single-digit numbers from the first step, add all
the digits in the odd places from right to left in the card number, sum the results
from the second and third steps, and if the result from the fourth step is divisible
by 10, the card number is valid; otherwise it is invalid.

Written by Deborah Barndt.
'''

```

```
24  # Main function to ask the user to enter a credit card number, then tell them if it
25  # is a valid or not valid credit card number and ask if they want to check another.
26  def main():
27      checkAgain = 'y'
28
29      while (checkAgain == 'y'):
30          number = int(input('Enter a credit card number between 13 and 16 digits: '))
31
32          if isValid(number):
33              print(number, 'is valid.')
34
35              # Ask user if they would like to check another credit card.
36              checkAgain = input('Would you like to check another credit card? (y/n) ')
37
38          else:
39              print(number, 'is not valid.')
40
41              # Ask user if they would like to check another credit card.
42              checkAgain = input('Would you like to check another credit card? (y/n) ')
43
44  # Return true if the card number is valid.
45  def isValid(number):
46      return getSize(number) >= 13 and getSize(number) <= 16 and \
47              (prefixMatched(number, 4) or prefixMatched(number, 5) or \
48               prefixMatched(number, 37) or prefixMatched(number, 6)) and \
49              (sumOfDoubleEvenPlace(number) + sumOfOddPlace(number)) % 10 == 0
50
51  # Get the result from Step 2.
52  def sumOfDoubleEvenPlace(number):
```

```
53      sum = 0
54
55      number = number // 10
56
57      while (number != 0):
58          sum += getDigit((number % 10) * 2)
59          number = number // 100
60      return sum
61
62  # Return this number if it is a single digit, otherwise, return the sum
63  # of the two digits.
64  def getDigit(number):
65      return number % 10 + (number // 10)
66
67  # Return sum of odd place digits in number.
68  def sumOfOddPlace(number):
69      sum = 0
70
71      while (number != 0):
72          sum += number % 10
73          number = number // 100
74      return sum
75
76  # Return true if the digit d is a prefix for number.
77  def prefixMatched(number, d):
78      return getPrefix(number, getSize(d)) == d
79
80  # Return the number of digits in d.
81  def getSize(d):
```

```
    length = 0

    while (d != 0):
        length += 1
        d = d // 10
    return length

# Return the first k number of digits from number. If the number of digits
# in number is less than k, return number.
def getPrefix(number, k):
    total = number

    for digit in range(getSize(number) - k):
        total //= 10
    return total

# Calling the main function to check the credit card number.
main()
```

Output Results:


Python 3.7.2 Shell
File Edit Shell Debug Options Window Help
RESTART: G:\ITMD 513 Open Source Programming Python\hw3\CreditCardNumberValidation.py
Enter a credit card number between 13 and 16 digits: 4388576018402626
4388576018402626 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 4388576018410707
4388576018410707 is valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 4975732895775
4975732895775 is valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 0517758204815
517758204815 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 1141670005087968
1141670005087968 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 8445553839409137
8445553839409137 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 4538759215534151
4538759215534151 is valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 5365526206474183
5365526206474183 is valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 7791829114667876
7791829114667876 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 3710293
3710293 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 5190990281925290
5190990281925290 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 3716820019271998
3716820019271998 is valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 3716820019271997
3716820019271997 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 8102966371298364
8102966371298364 is not valid.
Would you like to check another credit card? (y/n) y
Enter a credit card number between 13 and 16 digits: 6823119834248189
6823119834248189 is valid.
Would you like to check another credit card? (y/n) n
>>>