

A function is a mini-program that we can write separately from the main program, without having to think about the rest of the program while we write it. This allows us to reduce a complicated program into smaller, more manageable chunks, which reduces the overall complexity of our program.

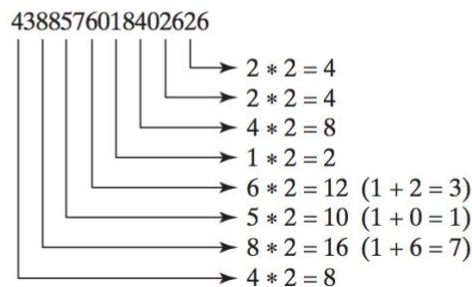
Write a program to validate a credit card number. Create all the functions listed below and implement them within your program.

(*Financial: credit card number validation*) Credit card numbers follow certain patterns: It must have between 13 and 16 digits, and the number must start with:

- 4 for Visa cards
- 5 for MasterCard credit cards
- 37 for American Express cards
- 6 for Discover cards

In 1954, Hans Luhn of IBM proposed an algorithm for validating credit card numbers. The algorithm is useful to determine whether a card number is entered correctly or whether a credit card is scanned correctly by a scanner. Credit card numbers are generated following this validity check, commonly known as the *Luhn check* or the *Mod 10 check*, which can be described as follows (for illustration, consider the card number 4388576018402626):

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.

$$4 + 4 + 8 + 2 + 3 + 1 + 7 + 8 = 37$$

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

$$6 + 6 + 0 + 8 + 0 + 7 + 8 + 3 = 38$$

4. Sum the results from Steps 2 and 3.

$$37 + 38 = 75$$

5. If the result from Step 4 is divisible by 10, the card number is valid; otherwise, it is invalid. For example, the number 4388576018402626 is invalid, but the number 4388576018410707 is valid.

```
# Return true if the card number is valid
def isValid(number):

# Get the result from Step 2
def sumOfDoubleEvenPlace(number):

# Return this number if it is a single digit, otherwise, return
# the sum of the two digits
def getDigit(number):

# Return sum of odd place digits in number
def sumOfOddPlace(number):

# Return true if the digit d is a prefix for number
def prefixMatched(number, d):

# Return the number of digits in d
def getSize(d):

# 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):
```

Write a Report Summary

Using Microsoft Word, answer the following questions. Please describe your responses. Do not simply answer yes or no - describe.

1. Did you complete your assignment and did it run without errors?
2. Did your program produce the correct result?
3. Did you test your program thoroughly?
4. How much time did you spend completing your assignment?

5. Did you write the program yourself? Did you get any help from anyone?
6. When you encountered obstacles to completing your program, how did you resolve the issues? Did you use Google to get help? Describe how Google was able or not able to assist you?
7. What did you learn from doing this assignment?
8. Any other information you would like to share with your instructor?

What to submit

1. Submit your program and your program output.
2. Submit your learning report summary.