```
# name: Justin Dang
# Student ID: 1148267
# Lab 03, Question 1

'''
determines wether a credit card is valid using luhn check
'''
def main():
    #user will input the credit card number as a string
    #call the function isValid() and print wether the credit card number is valid or not valid
    number = str(input('Enter a credit card number as a long integer:'))
    VALID = isValid(number)

    if VALID == True:
        print(number, 'is valid')
    elif VALID == False:
        print(number, 'is invalid')

def isValid(number)->(bool):
    #Returns true if the card number is valid
    #hint you will have to call function sumOfDoubleEvenPlace() and sumOfOddPlace()
    EVENSUM = sumOfDoubleEvenPlace(number)
    ODDSUM = sumOfOddPlace(number)
    TOTAL = ODDSUM + EVENSUM
    TOTAL%=10
    if TOTAL > 0:
        VALID = False
        return VALID
    elif TOTAL == 0:
        VALID = True
        return VALID

def sumOfDoubleEvenPlace(number:str)->(int):
    #Get the result from Step 2
    EVEN = ''
    for a in range(1, len(number), 2):
        EVEN += number[a]
    int(EVEN)
    EVENSUM = 0
    for x in EVEN:
        DOUBLE = int(x) * 2
        getDigit(DOUBLE)
        EVENSUM += DOUBLE
    return EVENSUM

def sumOfOddPlace(number:str)->(int):
    #Return sum of odd place digits in number
    ODD = ''
    for a in range(0, len(number), 2):
        ODD += number[a]
    int(ODD)
    ODDSUM = 0
    for x in ODD:
        ODDSUM += int(x)
    return ODDSUM

def getDigit(DOUBLE:int)->(int):
    #Return this number if it is single digit, otherwise return
    #the sum of the two digits
    if DOUBLE > 9:
        DIGIT1 = int(DOUBLE/10)
        DIGIT10 = DOUBLE%10
        DOUBLE = int(DIGIT1) + int(DIGIT10)
        return DOUBLE
    else:
        return DOUBLE
```

```python

if __name__ == '__main__':
    main()

## Output with test cases
##
## Test Case 1.
##
##Enter a credit card number as a long integer:4388576018410707
##4388576018410707 is valid

## Test Case 2.
##
##Enter a credit card number as a long integer:4388576018402626
##4388576018402626 is invalid

## Test Case 3.
##
##3788576018402626
##3788576018402626 is invalid

## Test Case 4.
##
##Enter a credit card number as a long integer:6011000990139424
##6011000990139424 is valid

## Test Case 4.
##
##Enter a credit card number as a long integer:5555555555554444
##5555555555554444 is invalid
```