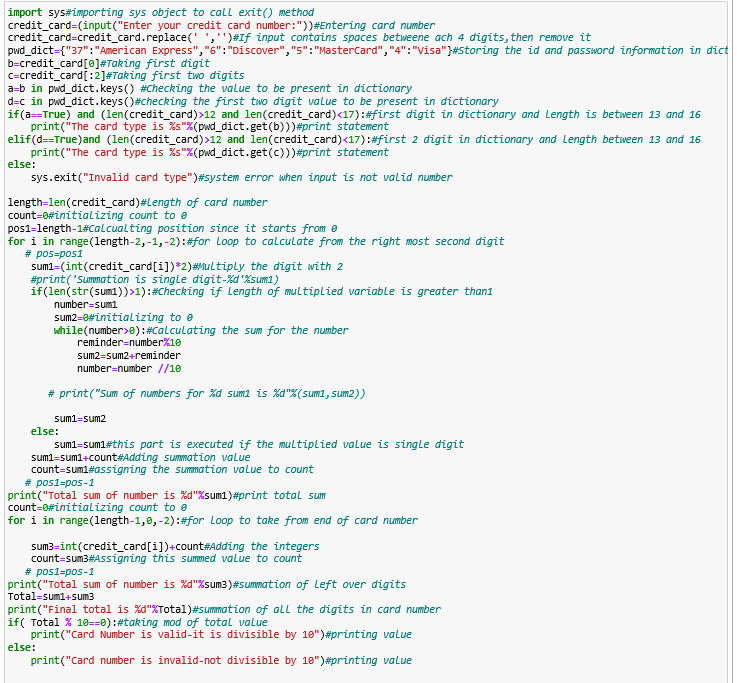
Program :



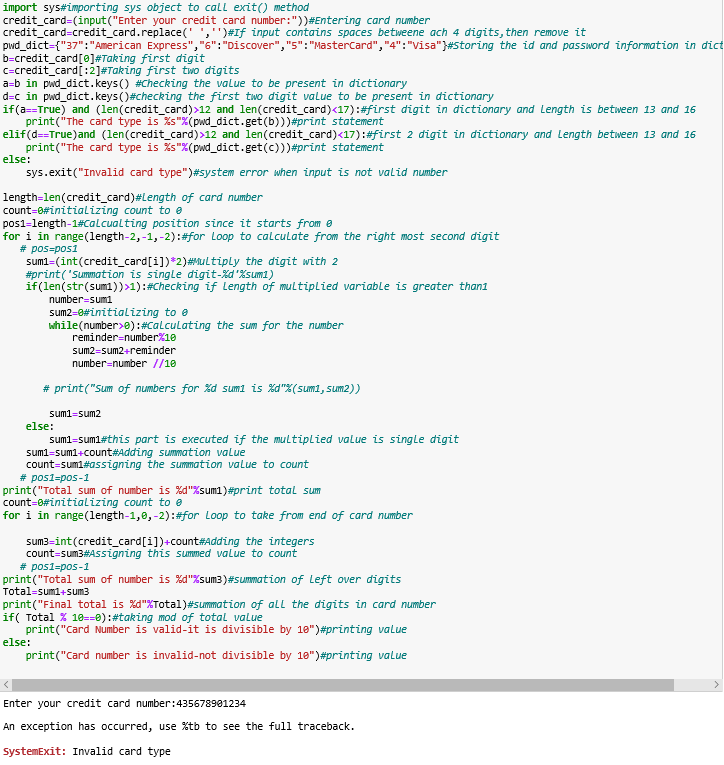
Output for invalid card where length of the cardnumber is between 13 and 16 but fails in cardnumber check process:



Output for valid card type:



Output for invalid card which is not present in dictionary and length is less than 13:



Questions:

1.Explain how placing your code into a separate module may benefit other python files involving credit card transactions.

We can place this credit card transaction check in a separate function so that all the companies which uses this check can call to that single function using credit card transaction check.

2.Investigate the link below. What are some other methods you can use in your project to verify a credit card transaction? Include at least three methods that may help validate user input of credit card data.

|  |
| --- |
| <https://www.codeproject.com/articles/2782/credit-card-validator-control-for-asp-net> |

1)Check cardnumber length

2.Check the prefix of the card number

3.Check card’s validity

|  |  |  |
| --- | --- | --- |
| **Card Type** | **Prefix** | **Number Length** |
| MasterCard | 51-55 | 16 |
| VISA | 4 | 13 or 16 |
| American Express | 34 or 37 | 15 |
| Diners Club/Carte Blanche | 300-305,36,38 | 14 |
| enRoute | 20,142,149 | 15 |
| Discover | 6011 | 16 |
| JCB | 3 | 16 |
| JCB | 21,311,800 | 15 |

As per the above table, other than the information present here, the input card numbers should be considered as invalid.

3.How would the use of regular expressions aid in reducing any bloated code to verify credit card data within an application? Give an example.

The first common regex we can place is to allow only numbers with 16 digits and no spaces or dashes allowed.If the card number has hyphens in between,we need to remove those characters prior to payment processing.

Example:^4\d{3}([\ \-]?)\d{4}\1\d{4}\1\d{4}$

This shows that starting form the 1st position ,match the digit ‘4’.From there,match 3 digits.And then match if any delimiter is present.Next match for 4 digits.Also,match on the same delimiter like before.Match on next 4 digits.And match last 4 digits.The final position is the end of string.

So,this regex would help in minimizing the number of lines used for credit card validation check.

4.Describe some ways to ultimately check your code to validate various card numbers for any faulty entry and / or invalid card numbers or even valid card numbers.

The code involves to check the first number and determine the card type.Also,with the credit card number we perform certain operations where we sum up the digits from last second number followed by alternate numbers.And,we do summation for the left out numbers from the end digit.On summing both the totals,if the number is divisible by 10,we make it as card number is valid.If this number is not divisible by 10,then the card number is invalid.

And,another way to check is,when the user inputs credit card number,if the length is less than 13,then the code stops execution and prints invalid card message.

5.Why would there be a reason to store any invalid numbers that were input by a particular user?

No,storing invalid numbers is not that essential.Instead storing the number of attempts the user has tried is important as only 3 attempts are allowed per user.