FAKE CREDIT CARD DETECTION CODE:

""# Fake Credit Card Detection System Using Luhn Algorithm

from datetime import datetime

def luhn\_algorithm(card\_number: str) -> bool:

"""Validates credit card number using Luhn's Algorithm."""

card\_number = card\_number.replace(' ', '')

if not card\_number.isdigit():

return False

total = 0

reverse\_digits = card\_number[::-1]

for idx, digit in enumerate(reverse\_digits):

n = int(digit)

if idx % 2 == 1:

n \*= 2

if n > 9:

n -= 9

total += n

return total % 10 == 0

def get\_card\_type(card\_number: str) -> str:

"""Identifies the credit card type based on its number."""

if card\_number.startswith('4'):

return 'Visa'

elif card\_number.startswith(('34', '37')):

return 'American Express'

elif card\_number.startswith(('51', '52', '53', '54', '55')):

return 'MasterCard'

elif card\_number.startswith('6011') or card\_number.startswith(('65',)):

return 'Discover'

else:

return 'Unknown'

def main():

print("== Credit Card Validator ==")

# User Inputs

card\_number = input("Enter the credit card number: ")

amount = input("Enter the transaction amount: ")

timestamp = input("Enter the transaction timestamp (YYYY-MM-DD HH:MM:SS): ")

# Validation

if luhn\_algorithm(card\_number):

card\_type = get\_card\_type(card\_number)

print("

=== Transaction Details ===")

print(f"Card Number: {card\_number}")

print(f"Card Type: {card\_type}")

print(f"Amount: ${amount}")

print(f"Timestamp: {timestamp}")

print("Status: Valid Card ✅")

else:

print("

Invalid credit card number ❌")

if \_\_name\_\_ == "\_\_main\_\_":

main()

""

**Brief Overview of the Code for Fake Credit Card Detection:**

The code is structured to:

1. **Validate Credit Card Numbers** using the **Luhn Algorithm**.
2. **Identifies the Card Type** (Visa, MasterCard, American Express, etc.).
3. **Capture Transaction Details** (Amount and Timestamp).
4. **Display the Validation Result** in a clear and structured format.

**Luhn Algorithm Validation – Detailed Explanation:**

The **Luhn Algorithm**, also known as the **Modulus 10** or **mod-10** algorithm, is a simple checksum formula used to validate various identification numbers, such as **CREDIT CARD NUMBERS.** It is primarily used to detect simple errors in card numbers, like mistyped digits.

**Steps of the Luhn Algorithm:**

1️. **Remove any spaces or non-digit characters** from the card number.  
2️. **Reverse the credit card number.**

* For example, 4539 4512 3456 7890 → 0987 6543 2154 9354.

3️. **Double every second digit** from the right (starting from index 1):

Original: 0 9 8 7 6 5 4 3 2 1 5 4 9 3 5 4

Double every second: 0 18 8 14 6 10 4 6 2 2 5 8 9 6 5 8

4️. **If the doubled value is greater than 9, subtract 9** from it:

Adjusted values: 0 9 8 5 6 1 4 6 2 2 5 8 9 6 5 8

5️. **Sum all the digits together**:

Sum = 0 + 9 + 8 + 5 + 6 + 1 + 4 + 6 + 2 + 2 + 5 + 8 + 9 + 6 + 5 + 8 = 84

6️. **Check if the sum is divisible by 10**:

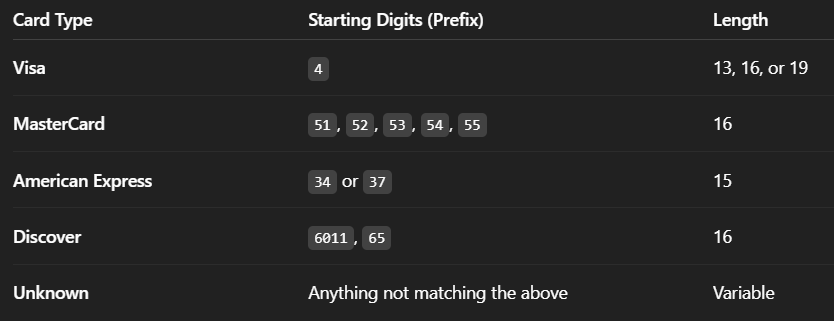
If 84 % 10 == 0, the card is **valid**.

If not, the card is **invalid**.

This Is how the **LUHN ALGORITHM VALIDATION** works.

Card Type Detection

The **Card Type Detection** in the code identifies the **issuer of the credit card** (like Visa, MasterCard, American Express, Discover) based on the **starting digits (prefixes)** of the card number.

Each card issuer uses specific prefixes that are unique to them.  
The code checks the beginning digits of the card number to identify the type: 

**Main Program Logic:**

def main():

print("== Credit Card Validator ==")

# User Inputs

card\_number = input("Enter the credit card number: ")

amount = input("Enter the transaction amount: ")

timestamp = input("Enter the transaction timestamp (YYYY-MM-DD HH:MM:SS): ")

# Validation

if luhn\_algorithm(card\_number):

card\_type = get\_card\_type(card\_number)

print("\n=== Transaction Details ===")

print(f"Card Number: {card\_number}")

print(f"Card Type: {card\_type}")

print(f"Amount: ${amount}")

print(f"Timestamp: {timestamp}")

print("Status: Valid Card ✅")

else:

print("\nInvalid credit card number ❌")

This is the main execution block.

It prompts the user to enter:

* **Credit Card Number**
* **Transaction Amount**
* **Timestamp** in the format: YYYY-MM-DD HH:MM:SS.
* If the **Luhn Algorithm validates** the card:
  + The card type is identified using get\_card\_type.
  + All the transaction details are displayed neatly.
  + The status is shown as **Valid Card ✅**.
* If the card is not valid:
  + It simply displays **Invalid credit card number ❌**.

**Summary of the Code Flow:**

1. **Input Collection** → Card number, amount, and timestamp.
2. **Luhn Validation** → Checks if the card is mathematically valid.
3. **Type Detection** → Identifies Visa, MasterCard, AmEx, etc.
4. **Display** → Prints the results with a status indicator.