**ATM Card Number Validation**

**1. Introduction**

This project checks whether an entered **ATM card number** is valid or not.  
It uses the **Luhn Algorithm** (a widely used checksum formula for card number validation).  
If the number passes the check, it is declared **Valid**; otherwise, it is **Invalid**.

**2. Process Explanation**

**Step 1: Taking Card Number Input**

The program first asks the user to enter their ATM card number.  
Example:

Enter your card number: 4539578763621486

**Step 2: Separating the Check Digit**

* The **last digit** of the card number is called the **check digit**.
* It is separated from the rest of the digits for the validation test.  
  Example:
* Card number: 4539578763621486
* Check digit: 6
* Remaining digits: 453957876362148

**Step 3: Reversing the Digits**

The remaining digits (without the check digit) are reversed.  
Example:

* Before reversal: 453957876362148
* After reversal: 841263678759354

**Step 4: Doubling Every Second Digit**

* Starting from the **first digit in the reversed list**, every second digit is doubled.
* If doubling makes a number greater than 9, **subtract 9** from it.

Example (on reversed digits 841263678759354):

* 8 → 16 → 16 - 9 = 7
* 4 → stays 4
* 1 → 2
* 2 → stays 2
* 6 → 12 → 12 - 9 = 3
* …and so on.

**Step 5: Summing the Digits**

All transformed digits are added together.  
Finally, the **check digit** (the last digit of the original card number) is also added to this sum.

**Step 6: Modulo Operation (Final Check)**

* The final sum is divided by **10**.
* If the remainder is **0**, the card number is valid.
* Otherwise, it is invalid.

Formula:

(total sum + check digit) % 10 == 0

**3. Example Run**

**Input:**

Enter your card number: 4539578763621486

**Process (Simplified):**

* Check digit: 6
* Remaining reversed digits processed with doubling and subtraction → sum calculated.
* Final sum + check digit → divisible by 10.

**Output:**

Valid Card: