# **CTF Challenge:Too many cards**

## **Challenge Overview**

In this Capture The Flag (CTF) challenge, participants are given a list of 100 encrypted credit card numbers. Among these, only one credit card number is valid according to the Luhn algorithm, which is a checksum formula used to validate credit card numbers.

### **Encryption Method**

Each credit card number is encrypted using two sequential encoding techniques:

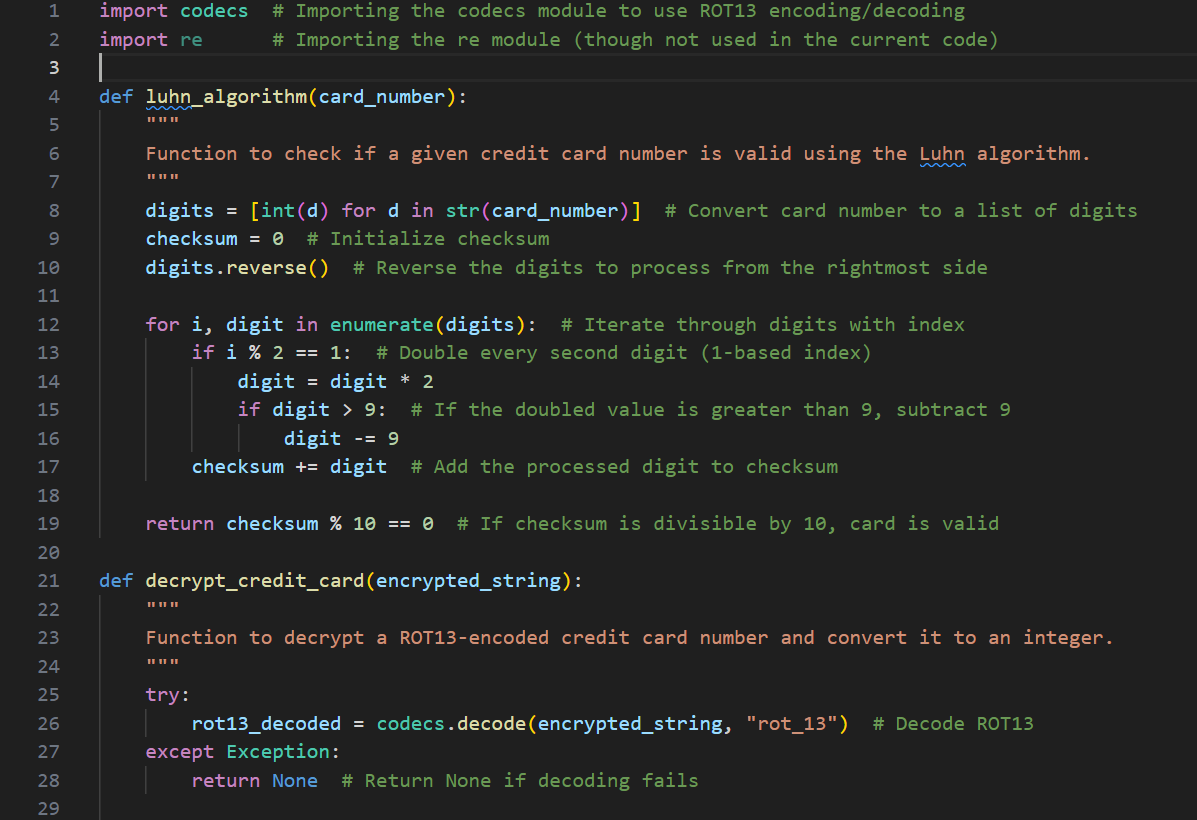
1. **Hexadecimal Encoding**: The original credit card number is first converted into a hexadecimal string.
2. **ROT13 Encoding**: The hexadecimal string is further obfuscated using the ROT13 cipher, which shifts letters by 13 places in the alphabet.

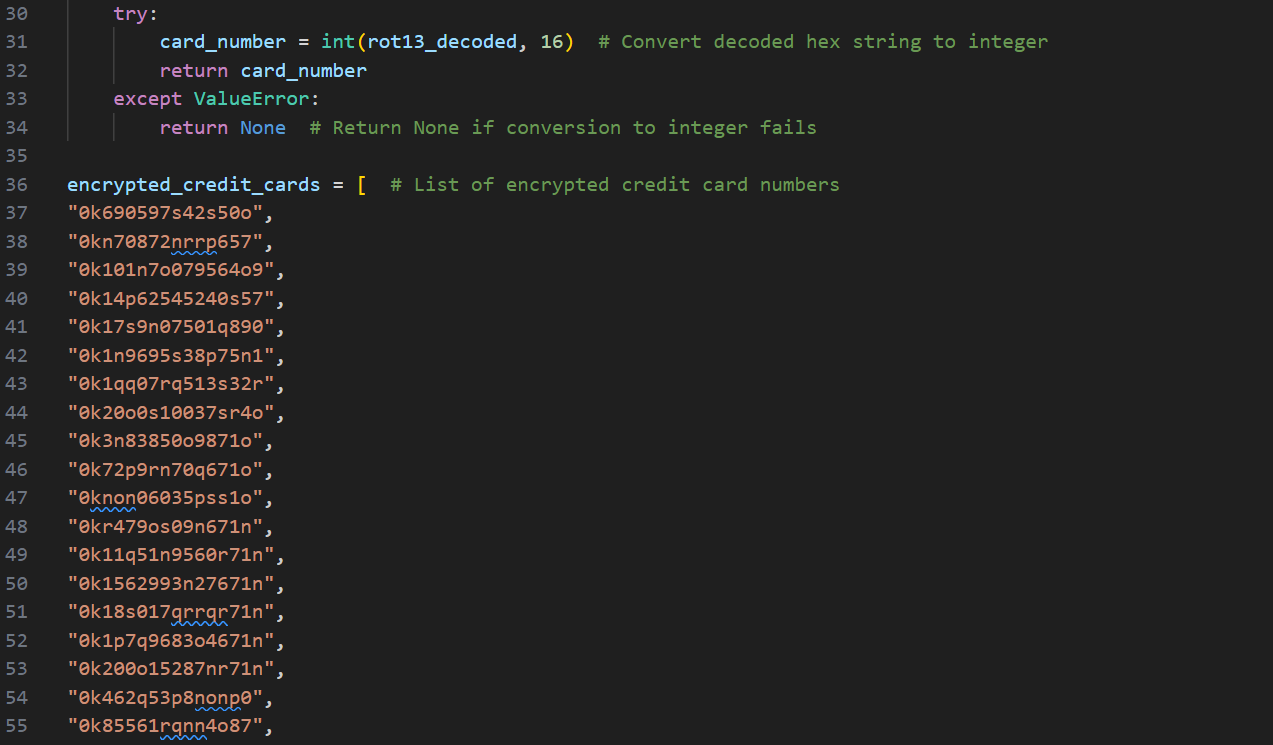
To find the correct answer, participants must:

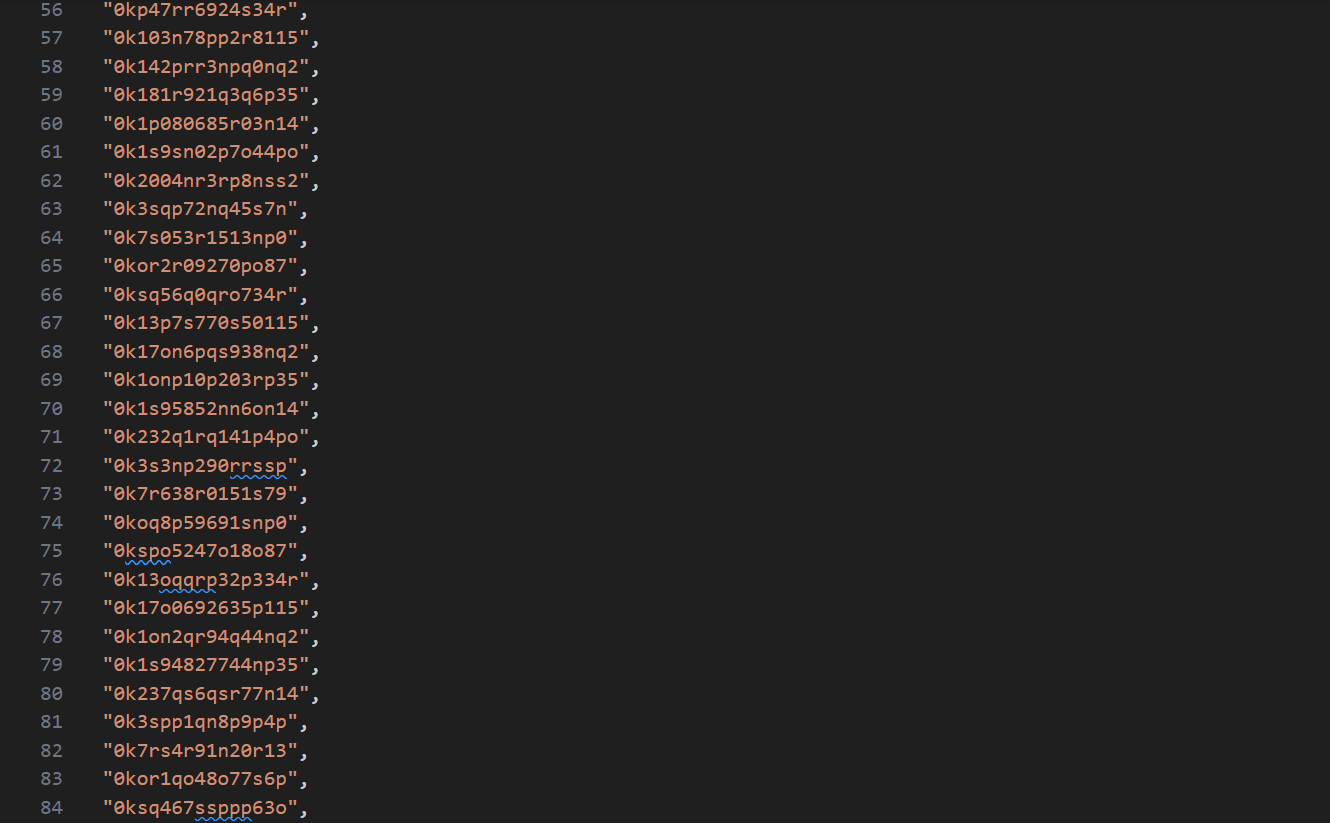
* Reverse the ROT13 encoding.
* Convert the hexadecimal back to an integer.
* Validate the decrypted number using the Luhn algorithm.

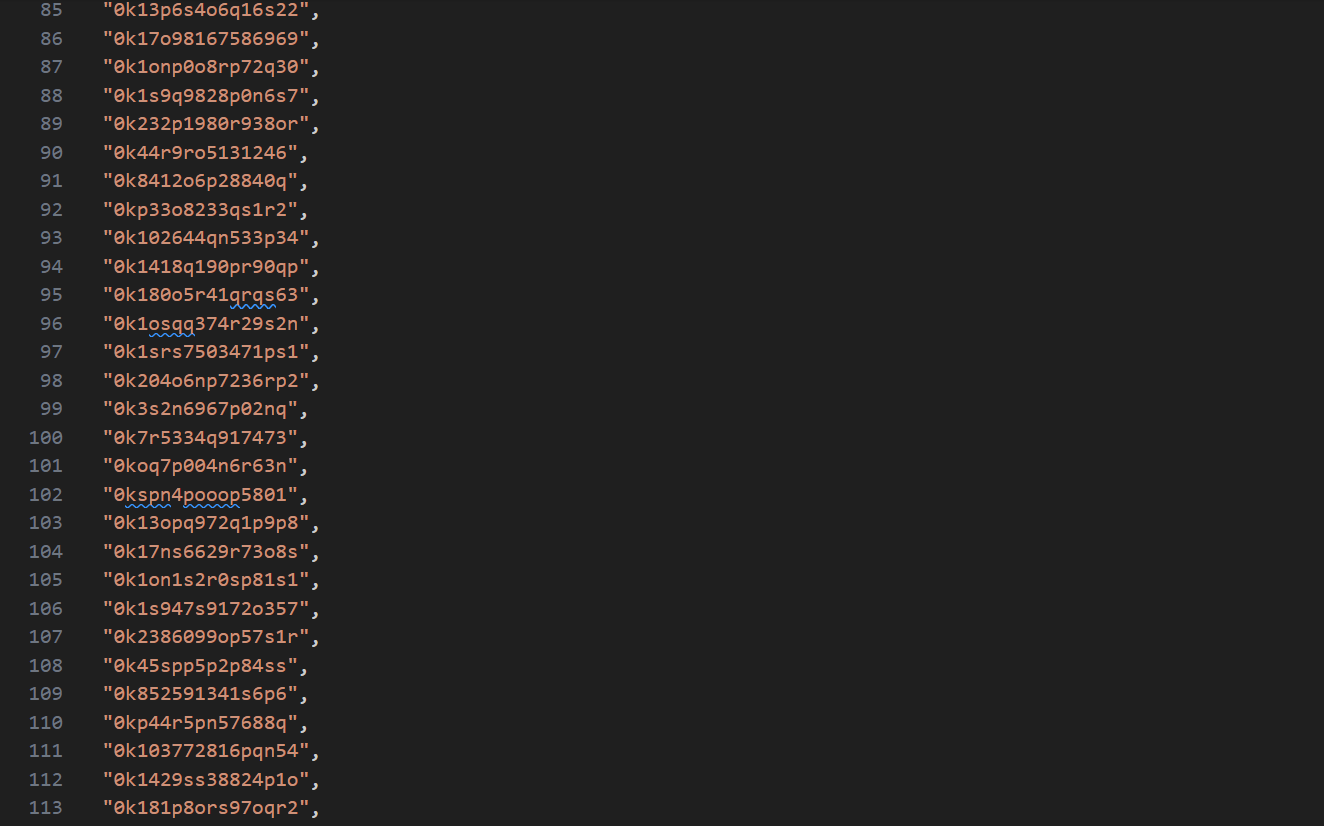
## **Solution Walkthrough**

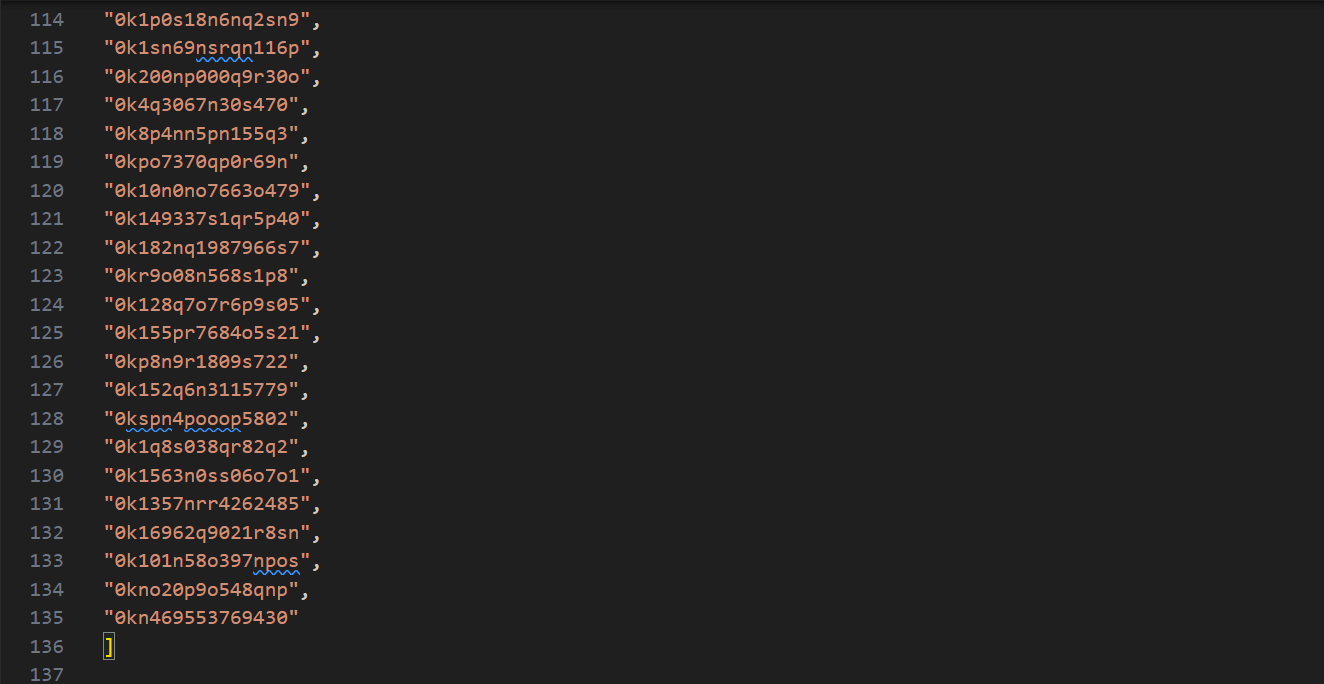
The following Python script decrypts the credit card numbers and checks for validity.

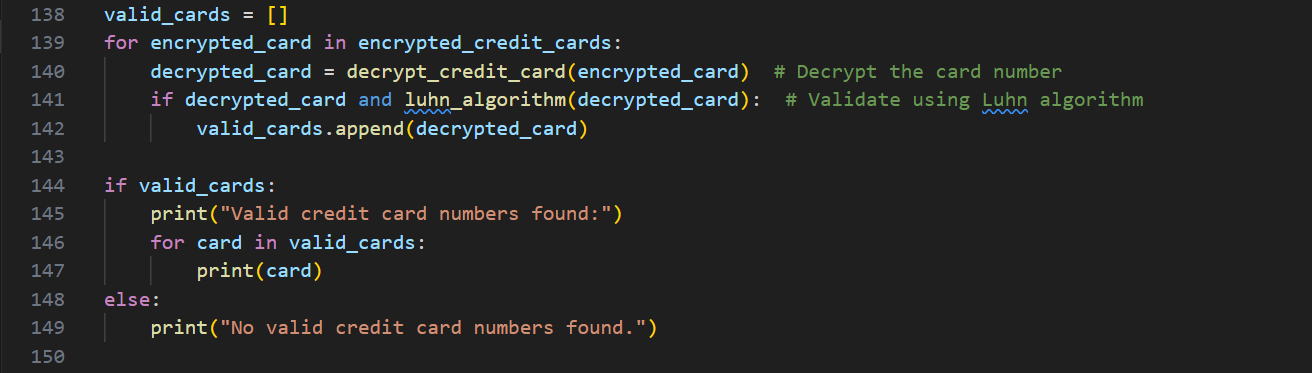










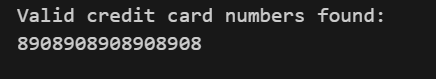


## **Explanation of the Code**

1. **Decrypting the Encrypted Credit Card Numbers:**
   * Each encrypted credit card number is decoded using the ROT13 cipher.
   * The decoded string is then converted from hexadecimal to an integer.
2. **Validating the Decrypted Credit Card Number:**
   * The decrypted number is checked using the **Luhn Algorithm** to determine its validity.
   * If the checksum satisfies the Luhn rule, the number is considered valid.
3. **Extracting the Valid Credit Card Number:**
   * The script iterates through all 100 encrypted numbers, decrypts each one, and validates them.
   * The valid card number is extracted and displayed.

## **The Answer**

After running the script, the valid credit card number is found to be:



To submit the answer, format it as

hacks{8908-9089-0890-8908}