Institute of Space Technology Islamabad

Discrete Structures

Project Report

Muhammad Bin Nasir	220201012
Aiza Shafqat	220201036

Project Title:

Application of Modular Arithematics especially Luhn’s Algorithm to authenticate Credit Cards such as Visa and MasterCard, UPC number and ISBN.

Introduction:

In our contemporary world, the pervasive challenges of cybersecurity dominate, prompting the implementation of various protocols to safeguard diverse organizations and processes. Major credit card manufacturers like MasterCard and Visa leverage Luhn's Algorithm to ascertain the validity of cards, thwarting the operation of invalid and counterfeit cards. This security measure is mirrored in the Universal Product Code (UPC), found on products in both barcode and numerical forms, and the International Standard Book Number (ISBN). These codes adhere to analogous protocols to uphold the security and integrity of associated products.

The imperative for a program that validates such codes arises from the necessity to confirm their authenticity and authenticate the corresponding products. Emphasis is placed on exploring the mathematical principles, specifically Luhn's Algorithm, employed in the validation process. This underscores the significance of mathematical ideas in fortifying the security and authentication mechanisms embedded within these essential codes.

What are we studying:

ISBN: A unique identifier for books, either 10 or 13 digits long, with a check digit ensuring accuracy.

UPC: A 12-digit barcode used for product identification, with the last digit as a check digit for error detection.

VISA: A 16-digit code identifying a Visa credit or debit card, starting with a 4 and validated using the Luhn algorithm.

MasterCard: A 16-digit code identifying a credit or debit card issued by MasterCard, starting with a 5 and verified using the Luhn algorithm.

Why do we need it:

The significance of this application lies in its assurance of accurate product and book identification, safeguarding financial transactions, and acting as a defense against potential fraud. Its absence could lead to frequent errors in inventory management, payment processing, and the recognition of books and products.

The necessity for such a program is underscored by the contemporary need for heightened security. In light of prevalent scams and frauds, establishing a system of protocols becomes imperative to uphold the integrity of products and shield them from malicious activities.

What tools are we using:

Utilizing tools from modular arithmetic, this program generates check digits for credit card numbers, ISBNs, and UPCs, establishing a secure and efficient authentication check digit system through modular classes. At its core is Luhn's Algorithm, employed to allocate weights to distinct code values, and the resulting check digit serves as a validator for the code's authenticity.

What program have we made:

We have developed a program in Python Language, with a Graphical User Interface using Tkinter Library that comes pre-included in Python. The program takes Credit Card number, UPC number, or ISBN and then checks if it is valid or not using Luhn’s Algorithm.

For Credit Cards:

1. The algorithm takes in a credit card number.
2. Then starting from right, it doubles every second digit, subtracting 9 if the result is greater than 9.
3. Then it adds all doubled and non-doubled values together.
4. Then, it takes the mod of resultant value on 10. If the remainder is zero, it is a valid card.

ISBN numbers there are of two types. ISBN-10 and ISBN-13.

For ISBN-10:

1. We assign weights to each index in reverse order from 10 to 1.

2. Then we add them together, and then mod it with 11.

3. If remainder is zero, it is likely valid,

For ISBN-13:

1. We assign weight of 3 to all even indices, and 1 to all odd indices, part the last index.

2. Then, we add all values together, and mod it with 10.

3. If the remainder is 0, then the checksum (last digit) should also be 0.

4. Otherwise, we subtract the remainder from 10, and the checksum should be that digit.

For UPC number:

1. We assign weights of 3 to all odd indices, and weight of 1 to all even indices up to index 11.

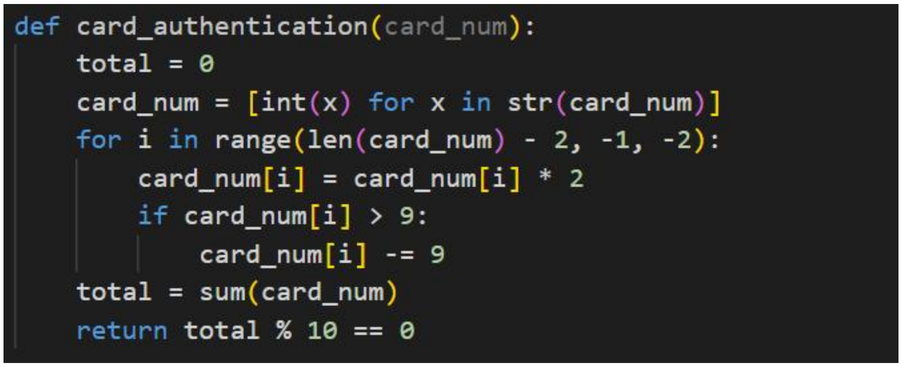
2. Then we add all the values together and take its mod with 11.

3. If the remainder is zero, then we check if last digit is also zero. If yes, then it is a valid number.

4. Else, we subtract the remainder from 10, and check if it is equal to last digit. If yes, then it is a valid number.

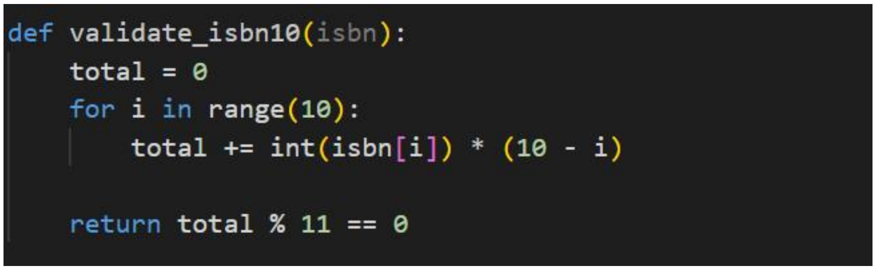
Code Snippets:

1. Credit Card Validation:



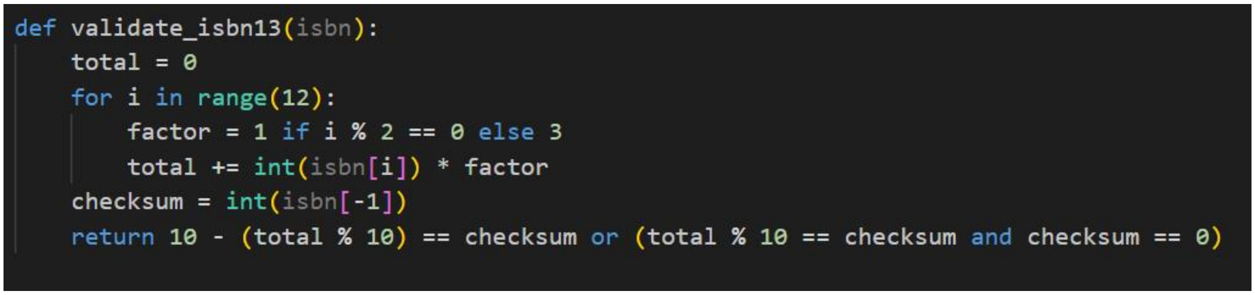
```python
def card_authentication(card_num):
    total = 0
    card_num = [int(x) for x in str(card_num)]
    for i in range(len(card_num) - 2, -1, -2):
        card_num[i] = card_num[i] * 2
        if card_num[i] > 9:
            card_num[i] -= 9
    total = sum(card_num)
    return total % 10 == 0
```

2. ISBN-10:



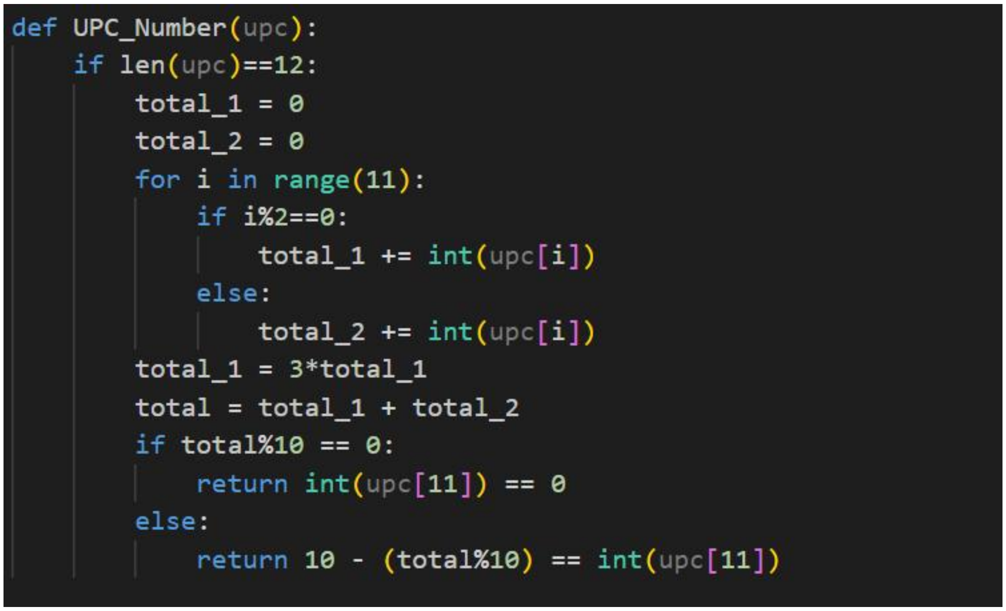
```python
def validate_isbn10(isbn):
    total = 0
    for i in range(10):
        total += int(isbn[i]) * (10 - i)

    return total % 11 == 0
```

3. ISBN-13:



```python
def validate_isbn13(isbn):
    total = 0
    for i in range(12):
        factor = 1 if i % 2 == 0 else 3
        total += int(isbn[i]) * factor
    checksum = int(isbn[-1])
    return 10 - (total % 10) == checksum or (total % 10 == checksum and checksum == 0)
```

4. UPC Number:



```
def UPC_Number(upc):
    if len(upc)==12:
        total_1 = 0
        total_2 = 0
        for i in range(11):
            if i%2==0:
                total_1 += int(upc[i])
            else:
                total_2 += int(upc[i])
        total_1 = 3*total_1
        total = total_1 + total_2
        if total%10 == 0:
            return int(upc[11]) == 0
        else:
            return 10 - (total%10) == int(upc[11])
```

Conclusion:

This project exemplifies the practical application of discrete structures, notably focusing on modular arithmetic, to elevate authentication procedures for credit card numbers, UPCs, and ISBNs. Through the implementation of the Luhn algorithm, the code and presentation contribute significantly to a comprehensive understanding of how mathematical concepts fortify digital security in various applications.

By delving into the intricacies of the Luhn algorithm, the project sheds light on its versatile utility in bolstering authentication mechanisms for credit cards, UPCs, and ISBNs. The systematic examination of these applications not only enhances the project's educational value but also underscores the algorithm's paramount role in ensuring robust and dependable security measures. As a testament to the symbiosis of mathematical principles and real-world challenges, this project provides valuable insights into the intersection of theoretical concepts and their pragmatic implementation in the realm of digital security.

References:

1. https://slideplayer.com/slide/4688347/
2. https://bookscouter.com/tips/isbn-10-vs-isbn-13-whats-the-difference/#:~:text=Books%20published%20later%20than%20January,check%20digits%20(last%20numbers)
3. https://www.geeksforgeeks.org/program-check-isbn/
4. https://barcodes.pk/upc-barcodes/
5. https://www.gs1us.org/upcs-barcodes-prefixes/guide-to-upcs#:~:text=A%20UPC%20code%2C%20which%20stands,with%20these%20lines%20and%20digits
6. https://en.wikipedia.org/wiki/Luhn_algorithm
7. https://realpython.com/python-gui-tkinter/
8. https://chat.openai.com/c/f798b9c1-880c-458f-ba33-dd15f670b6cd
9. https://www.educationise.com/post/what-is-discrete-mathematics#:~:text=Discrete%20Math%20Definition&text=It%20plays%20a%20vital%20role,problem-solving%20and%20analytical%20thinking