# Credit Card Validation Using Lhun's Algorithm

Lauvindra Raj

May 2021

## Introduction

The Luhn's algorithm, also known as the modulus 10 or mod 10 algorithm, is a basic checksum formula that is used to validate a wide range of identification numbers, including credit card numbers, IMEI numbers, and some of insurance numbers. The algorithm is used by most credit cards and many government identification numbers to differentiate legitimate numbers from mistyped or otherwise incorrect numbers. It was created to guard against unintentional errors rather than deliberate attacks.

## Procedure

These are the procedures to validate a credit card. These can be done manually for a credit card, we can use this algorithm to write a code using Python 3 based on these procedures for a more fast and easy validation.
Before we begin our validation the numbers should be listed down and numbered using index.

1. Find the sum of odd index values of the credit card.

2. Find the number of even numbers.

3. Calculate the double of even numbers.

4. Find sum of even numbers.

5. Find sum of odd and even numbers if add up to the multiples of 10 then the credit card is valid.

## How it works

We input a card_no = "5610591081018250" and follow the procedure where it uses Luhn’s algorithm to find out the credit card is valid or not.

## Strengths and weaknesses

Any single-digit error, as well as almost all transpositions of adjacent digits, will be detected by the Luhn’s algorithm. However, it will not detect the two-digit sequence 09 to 90 being transposed (or vice versa). It can detect the majority of potential twin errors. Zero-padding the beginning of a string of numbers has no effect on the calculation since the algorithm works from right to left and zero digits only affect the result if they cause a change in place. As a result, systems that pad to a particular number of digits (for example, by converting 1234 to 0001234) may perform Luhn validation either before or after the padding and get the same result.