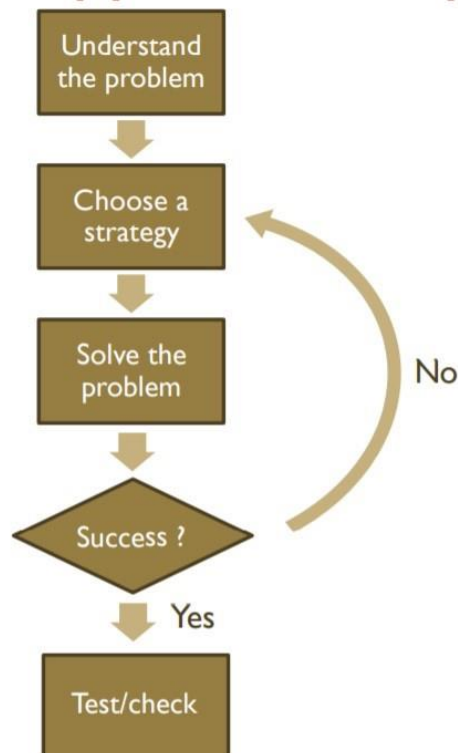


# Algorithm for the simulation

To tackle this problem of validating the credit card numbers. We are going to approach the problem using the flowchart of how to approach problems:

## How do we approach the problem?



### Understanding the problem:

The problem is that someone wants to simulate a program that can detect whether a provided credit card number is a valid number or not. Doing so by following the Mod 10 check, stated in the assignment.

### Choose a strategy:

We choose to use the solution strategy “The divide the problem into several subproblems”. Using this will help us simplify the problem and dividing the bigger problem into little smaller problems. When dividing the bigger problem into sufficient amount of smaller problems, go over to solving those little problems one by one. When completing this, we can say that we have solved our bigger problem.

The bigger problem has to be divided into subproblems to get the simulation program.

**The main problem:** create an algorithm that validate credit card numbers according to the criteria of credit card number combinations from companies such as VISA, Mastercard, American Express and Discover cards.

## Using the principle of inclusion and exclusion

Defining the principle of inclusion and exclusion:

This is a counting technique that computes the number of elements that satisfy at least one of several properties while guaranteeing that elements satisfying more than one property are not counted twice

- **Exclusion boundary 1:**  
Length of credit card number should be between 13 and 16 digits.  
The boundary includes 13 and 16.
- **Exclusion boundary 2:**  
Checking whether the credit card number starts with a valid number.
- **Exclusion boundary 3:**  
Checking whether the modulus at the end is equal to zero or not.
- **Inclusion boundary 1:**  
Checking only the even digits from right to left.
- **Inclusion boundary 2:**  
Checking only the odd digits from right to left.

### Simplifying the problem:

When the steps above are solved, what happens then? The end goal is to have a result and divide that by ten. When the division gives back a remainder of zero, the number is valid, otherwise the number is invalid.

When IB1 (inclusion boundary 1) is solved, every digit has to be multiplied by two. If the result of the multiplication is a two digit number, we add those two numbers up. Finally, when we have all of the even digits, we add them up to get a result for the even indices.

When IB2 (inclusion boundary 2) is solved, every digit has to be added to the previous one. Only this time, no multiplication happens. So, we get the result faster compared to IB1. Now, have a result for the odd indices.

We add the result of the even and odd indices. Now, we have the result that goes to EB3 (exclusion boundary 3).

To solve EB3, we take the modulus of the sum of the even and odd digits by ten and see whether there is a remainder of zero. If there is a remainder of zero, the credit card number is valid. If not, the number is invalid.

**Solving the problem:**

This what we have created using the Luhn's algorithm.

**The algorithm:**

1. Ask user for credit card number
2. Check whether the credit card number is between 13 and 16 digits
3. Check with what number the credit card number begins:
4. 4 for Visa cards
5. 5 for Master cards
6. 37 for American Express cards
7. 6 for Discover cards
8. Double every digit in the even places from right to left.
9. If doubling of a digit results in a two-digit number, add up the two digits to get a single-digit number.
10. Now add all single-digit numbers from Step 8.
11. Add all digits in the odd places from right to left in the card number.
12. Sum the results from Step 9 and Step 10.
13. Take the modulus of the result in step 11 by 10 and if the remainder is 0 then the credit card number is correct  
If the remainder is not 0 then the credit card number is invalid.

**Success?**

We were able to solve the problem by analysing which steps there had to be taken. The algorithm is what we got.

**Test/check:**

When following this algorithm, the credit card number is being checked. Following the steps will lead to an answer whether the credit card number is valid or not. So, our general algorithm provides a solution when a user submit a credit card number. This solution is that either the number is valid or invalid.