

# COMP 5481

## LAB 1 – Credit Card Validation

Most credit card numbers are encoded with a "Check Digit". A check digit is a digit added to a number (either at the end or the beginning) that validates the authenticity of the number. A simple algorithm is applied to the other digits of the number which yields the check digit. By running the algorithm, and comparing the check digit you get from the algorithm with the check digit encoded with the credit card number, you can verify that they make a valid combination.

### The Problem:

Write a java program to validate a given credit card number. For this problem, you should make use of the LUHN Formula (Mod 10) for validating credit card numbers. The algorithm has following steps:

Step 1: Double the value of alternate digits of the credit card number beginning with the second digit from the right (the rightmost digit is the check digit.)

Step 2: Add the individual digits comprising the products obtained in Step 1 to each of the unaffected digits in the original number.

Step 3: The total obtained in Step 2 must be a number ending in zero (30, 40, 50, etc.) for the account number to be validated. The total mod 10 = 0.

For example, to validate the credit card account number 49927398716:

Step 1:	4	9	9	2	7	3	9	8	7	1	6
		x2		x2		x2		x2		x2	
	—	—	—	—	—	—	—	—	—	—	—
		18		4		6		16		2	
Step 2:	4 +(1+8)+ 9 + (4) + 7 + (6) + 9 +(1+6) + 7 + (2) + 6										
Step 3:	Sum = 70: Card number is valid because the 70/10 yields no remainder.										

### The Input:

The input will comprise of one line of integer data which is a credit card number. The credit card number will have less than 20 digits.

### The Output:

If the credit card number is valid, output the word "VALID". If the credit card number is not valid, output the word "INVALID" followed by a space and the correct check digit, which is the right-most digit, which would make the credit card number valid.

No	Sample Input	Sample Output
1	49927398716	VALID
2	513467882134	INVALID 2
3	432876126	VALID