Problem name: Credit
Input File: CreditIn.txt

Although Zack's on-line store sales are booming, he's very unhappy. Many of the purchases are being billed to invalid credit cards. As a result, the more items Zack sells the further he goes into debt. Desperate, Zack has decided to hire you and your team to develop a credit card validation program.

You have decided to use a checksum scheme to validate the card numbers. Under this scheme credit card numbers must be 16 digits in length. Since actual credit card numbers varying in length from 13 digits to 16 digits, numbers with less than 16 digits will be zero-filled from the left to create a sixteen digit number. Then, starting from the left most digit, each digit of the credit card is multiplied by a weighting factor alternating between 2 and 1, beginning with a weighting factor of 2. If the number multiplied by the weighting factor results in a 2-digit number, each digit is added to the sum. If the final sum (the checksum) is a multiple of 10, the card is valid. Otherwise, it is invalid.

Example using the *invalid* card number: 99 0011 0012 0034

0	0	9	9	0	0	1	1	0	0	1	2	0	0	3	4	
2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	
0	+0	+1+8	+9	+0	+0	+2	+1	+0	+0	+2	+2	+0	+0	+6	+4	=35

Inputs:
A credit card number varying in length from 13 to 16 digits with a space between 4 digit groupings.

Outputs:
The zero-filled (if less than 16 digits long) credit card number, the checksum, and a status of "Valid" or "Invalid" depending on the success or failure of the checksum.

Sample Input
99 0011 0012 0034
5499 0011 0012 0034

Sample Output
Card Number: 5499 0011 0012 0034
Checksum: 35
Status: Valid

Card Number: 0099 0011 0012 0034
Checksum: 40
Status: Invalid