Exercise 2

We want to write a program that checks whether a given number is a correct credit card number or not. This task will be solved by composing several functions each with a specific role that is described in what follows:

1) First we want to transform a given integer number N into a list of integer where each digit of N is represented by an element of the list. For example, with N=23045, the list is [2,3,0,4,5]. For the next steps is useful to build the list in the inverse order, i.e., [5,4,0,3,2].
2) the list (assumed to be inverted) is then manipulated as follows: every second element of the list should be doubled and, in case the result is larger than 9, 9 should be subtracted to the value. In the above example [5,4,0,3,2] would become [5,8,0,6,2]. Starting with the list [1,8,3,7,8], we would obtain [1,7,3,5,8]
3) Finally, the elements of the list obtained after step (2) should be all summed up and then the resulting value should be a multiple of 10. If this is true then the original number is a correct credit card number, otherwise is not. Thus N=23045 is not a valid credit card number, whereas 40402 is a good number.
   The function executing step (3) should be called validate with the following type:
   validate :: Num a => a -> Bool, the result indicates whether the input number is a good credit card number or not.