Class Exercise:

Generators

You have been hired by a shipping company to develop a Python class that generates unique tracking numbers for packages. The tracking numbers should consist of a prefix, a sequential number, and a checksum digit. To achieve this, you need to implement a generator function inside the class that generates these tracking numbers.

Your task is to create a class called **TrackingNumberGenerator** with the following methods:

1. **\_\_init\_\_(self, prefix):** This method initializes the **TrackingNumberGenerator** class with the given prefix. The prefix is a string that will be included at the beginning of each generated tracking number.
2. **generate\_checksum(self, number):** This method takes a number as input and calculates the checksum digit based on a specific algorithm. The algorithm for calculating the checksum digit is as follows: Starting from the rightmost digit, multiply every second digit by 2. If the multiplication results in a number greater than 9, subtract 9 from it. Sum up all the digits, including the products and the unaffected digits. The checksum digit is the smallest number that, when added to the sum, makes it a multiple of 10.
3. **generate\_tracking\_number(self):** This method is a generator function that yields the next unique tracking number. Each generated number should have the prefix, a sequential number starting from 1, and the checksum digit.

class TrackingNumberGenerator:

    def \_\_init\_\_(self, prefix):

# Your code here

    def generate\_checksum(self, number):

# Your code here

    def generate\_tracking\_number(self):

# Your code here

# Usage example

generator = TrackingNumberGenerator("ABC")

print(generator.generate\_tracking\_number())  # Output: "ABC1"

print(generator.generate\_tracking\_number())  # Output: "ABC2"

print(generator.generate\_tracking\_number())  # Output: "ABC3"

Expected output:

ABC1

ABC2

ABC3