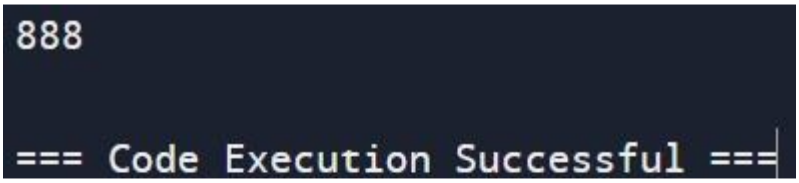
7. Max Difference You Can Get From Changing an Integer You are given an integer num. You will apply the following steps exactly two times: ● Pick a digit x (0 <= x <= 9). ● Pick another digit y (0 <= y <= 9). The digit y can be equal to x. ● Replace all the occurrences of x in the decimal representation of num by y. ● The new integer cannot have any leading zeros, also the new integer cannot be 0. Let a and b be the results of applying the operations to num the first and second times, respectively. Return the max difference between a and b. Example 1: Input: num = 555 Output: 888 Explanation: The first time pick x = 5 and y = 9 and store the new integer in a. The second time pick x = 5 and y = 1 and store the new integer in b. We have now a = 999 and b = 111 and max difference = 888

PROGRAM:-

```
def maxDiff(num):
    num_str = str(num)

    # To get the maximum possible value, replace the first non-9 digit with 9
    max_num_str = num_str
    for digit in num_str:
        if digit != '9':
            max_num_str = num_str.replace(digit, '9')
            break
    max_num = int(max_num_str)

    # To get the minimum possible value, replace the first non-1 digit with 1
    min_num_str = num_str
    if num_str[0] != '1':
        min_num_str = num_str.replace(num_str[0], '1')
    else:
        for digit in num_str[1:]:
            if digit != '0' and digit != '1':
                min_num_str = num_str.replace(digit, '0')
                break
    min_num = int(min_num_str)

    return max_num - min_num

# Example usage
num = 555
print(maxDiff(num))  # Output: 888
```

OUTPUT:-



```
888

=== Code Execution Successful ===
```

TIME COMPLEXITY:-O(n)