

37) 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 ( $0 \leq x \leq 9$ ). • Pick another digit y ( $0 \leq y \leq 9$ ). 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.

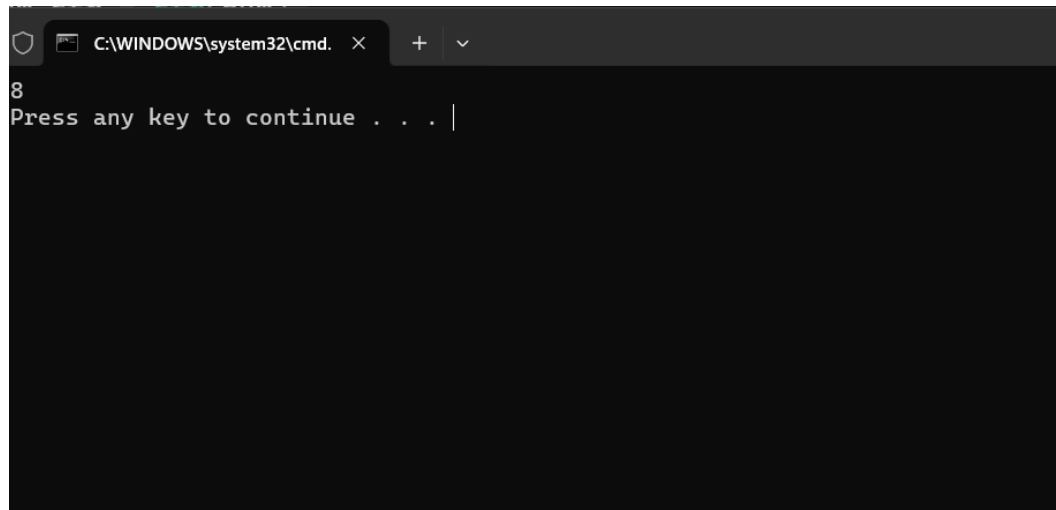
CODE:

```
def maxDiff(num: int) -> int:
    num_str = str(num)
    max_diff = 0

    for i, digit in enumerate(num_str):
        if digit != '9':
            new_num_str = num_str.replace(digit, '9')
            max_diff = max(max_diff, int(new_num_str) - num)
            break
    if num_str[0] != '1':
        new_num_str = num_str.replace(num_str[0], '1')
        max_diff = max(max_diff, num - int(new_num_str))

    return max_diff
print(maxDiff(9))
```

OUTPUT:

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\WINDOWS\system32\cmd.' and standard window controls. The command prompt displays the number '8' on the first line, followed by the text 'Press any key to continue . . . |' on the second line, with a vertical cursor at the end.

TIME COMPLEXITY :  $O(d)$