COMPUTER SOCIETY OF INDIA (CSI)
COEP TECH STUDENT CHAPTER

PRESENTS

CODE QUEST 5.0

EDITIORIAL

LIBRARY BOOK HUNT

EXPLANATION

#Solution Approach
The problem is a variant of the famous Bleatrix Trotter's insomnia problem. The main goal is to keep generating multiples of N until we have encountered all digits from 0 to 9.

Steps to Solve the Problem:

1.Handle edge case: If N = 0, immediately return "UNSOLVABLE" since no multiples will ever contain digits.

2.Use a set to track seen digits: Create an empty set seen_digits to store encountered digits.

3.Iterate over multiples of N:
 -Multiply N by increasing integers (1, 2, 3, ...).
 -Convert each multiple to a string and add its digits to seen_digits.
 -If seen_digits contains all digits from 0 to 9, return the last multiple.

Output the result for each test case.

EXPLANATION

#Explanation of code
1.Base Case: If N = 0, return "UNSOLVABLE" immediately.

2.Tracking Digits: Use a set seen_digits to keep track of encountered digits.

3.Looping through Multiples:
 -Generate the next multiple.
 -Extract digits and update seen_digits.
 -Stop when all 10 digits (0-9) are seen.
4.Return the last multiple where all digits were seen.

#Complexity Analysis
Time Complexity: O(d) per test case, where d is the number of multiples needed to collect all digits. Since digits are at most 6 digits long (10^6), this is efficient.

Space Complexity: O(1), since we only store a small set of digits.

EXPLANATION

Edge Cases Considered

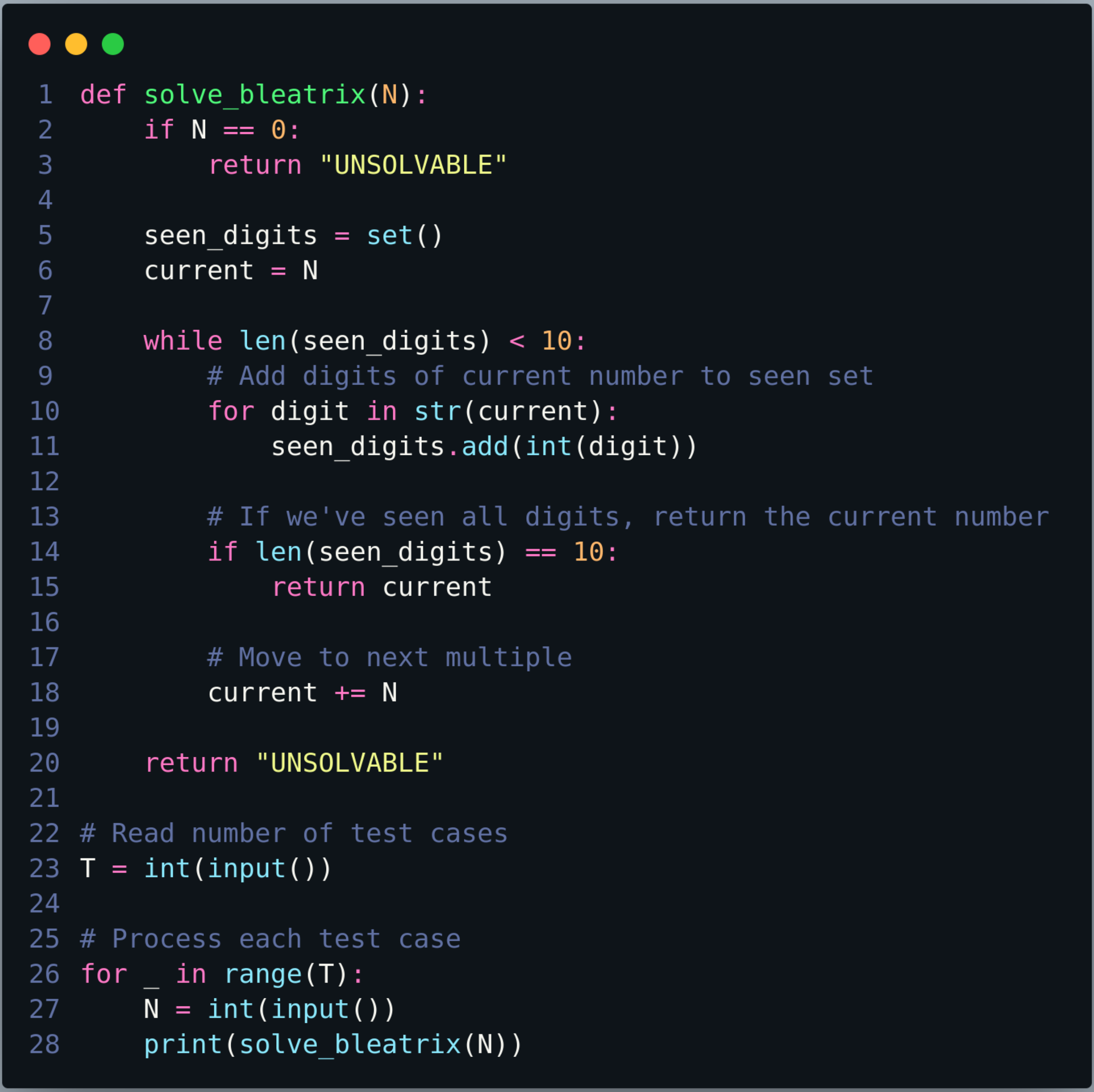
N = 0 (Always outputs "UNSOLVABLE").

N = 1 (Smallest valid number, quickly reaches all digits).

N = 1692 (Verifies correct stopping condition).

Large numbers (N near 10^6) to test efficiency.

SOLUTION



```python
def solve_bleatrix(N):
    if N == 0:
        return "UNSOLVABLE"

    seen_digits = set()
    current = N

    while len(seen_digits) < 10:
        # Add digits of current number to seen set
        for digit in str(current):
            seen_digits.add(int(digit))

        # If we've seen all digits, return the current number
        if len(seen_digits) == 10:
            return current

        # Move to next multiple
        current += N

    return "UNSOLVABLE"

# Read number of test cases
T = int(input())

# Process each test case
for _ in range(T):
    N = int(input())
    print(solve_bleatrix(N))
```