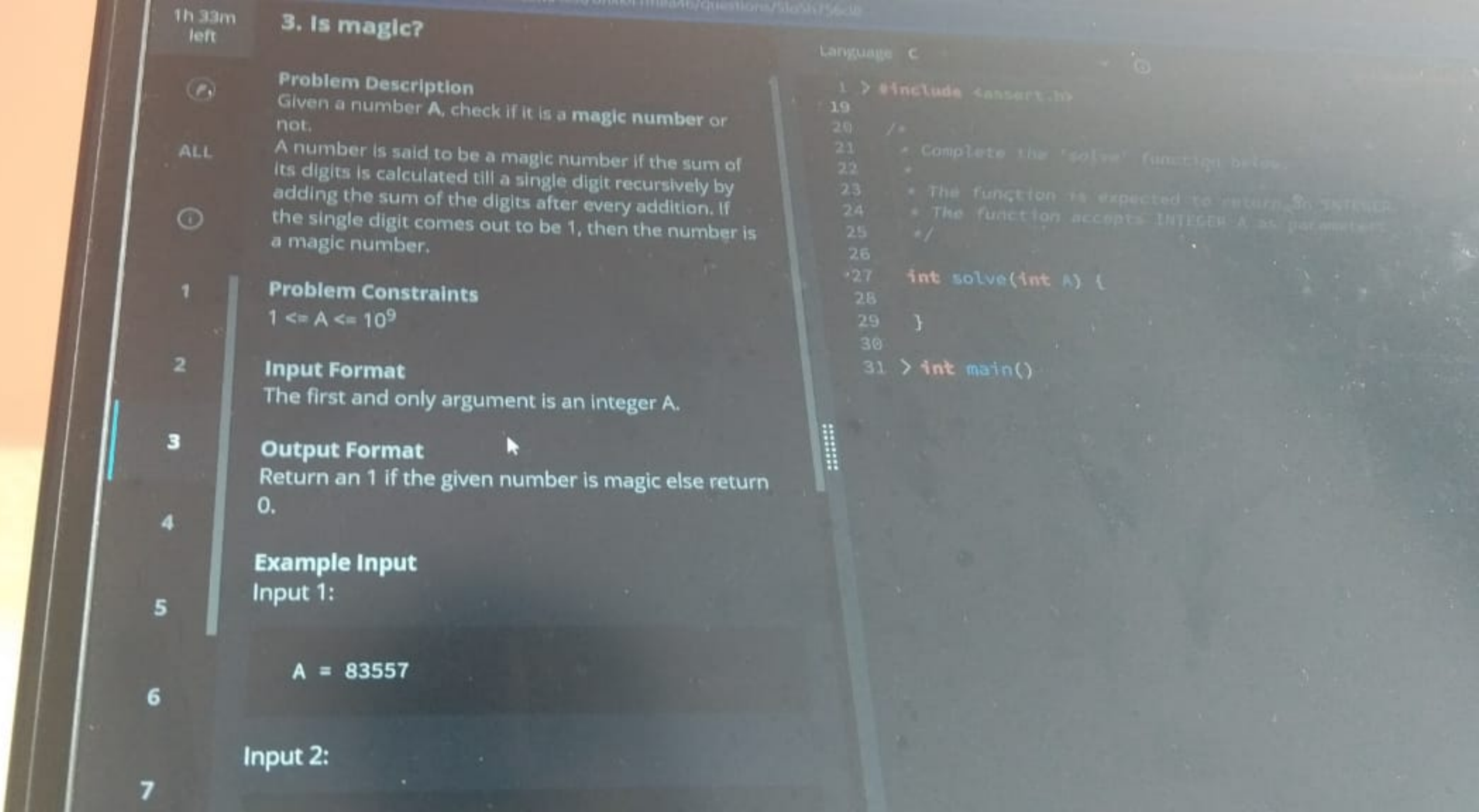


3. Is magic?

Problem Description

Given a number **A**, check if it is a **magic number** or not.

A number is said to be a magic number if the sum of its digits is calculated till a single digit recursively by adding the sum of the digits after every addition. If the single digit comes out to be 1, then the number is a magic number.

Problem Constraints

$1 <= A <= 10^9$

Input Format

The first and only argument is an integer A.

Output Format

Return an 1 if the given number is magic else return 0.

Example Input

Input 1:

```
A = 83557
```

Input 2:

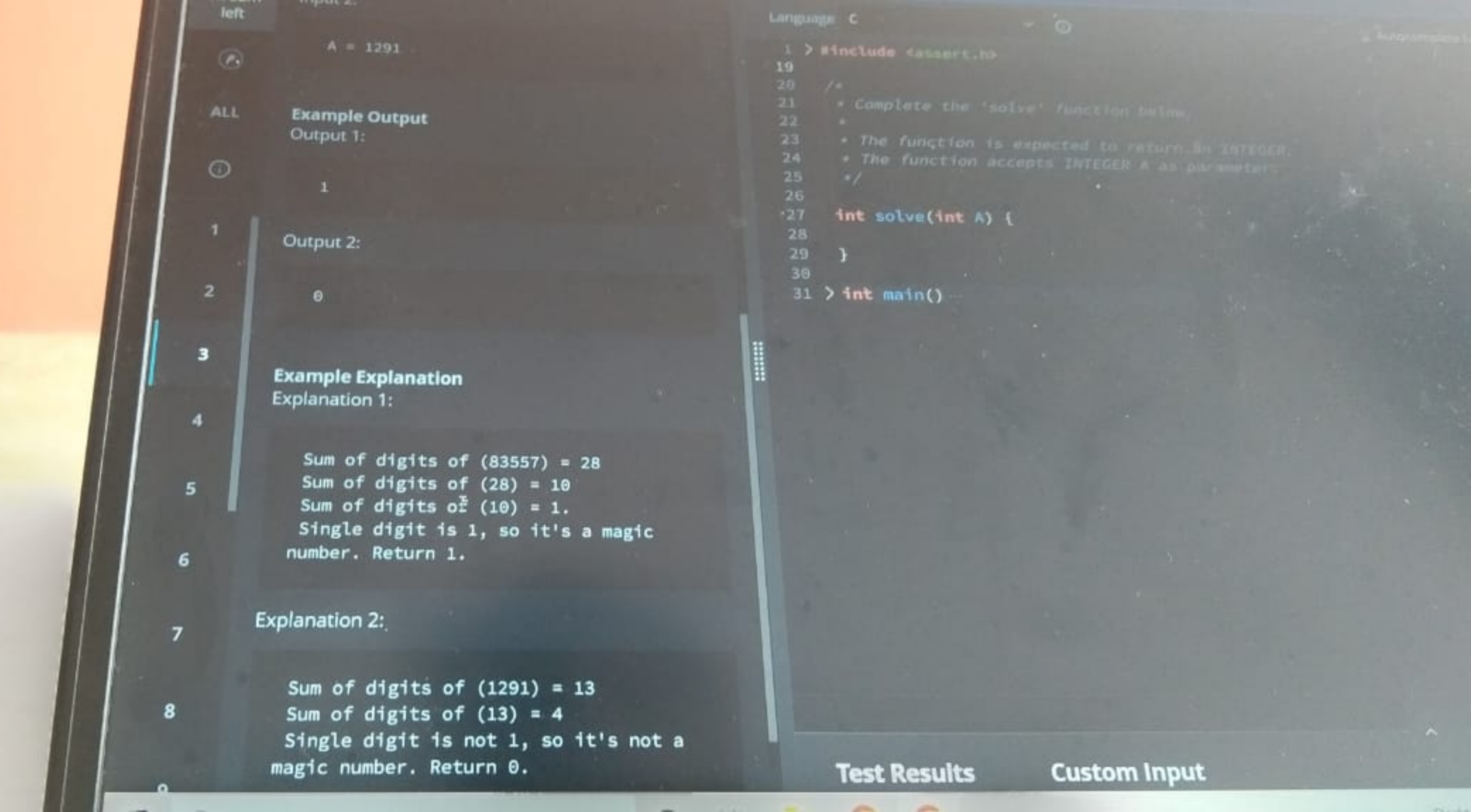
```
> #include <assert.h>

/*
 * Complete the 'solve' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER A as parameter.
 */

int solve(int A) {

}

> int main()
```



```
A = 1291
```

Example Output

Output 1:

```
1
```

Output 2:

```
0
```

Example Explanation

Explanation 1:

```
Sum of digits of (83557) = 28
Sum of digits of (28) = 10
Sum of digits of (10) = 1.
Single digit is 1, so it's a magic
number. Return 1.
```

Explanation 2:

```
Sum of digits of (1291) = 13
Sum of digits of (13) = 4
Single digit is not 1, so it's not a
magic number. Return 0.
```

Language C

```c
> #include <assert.h>

/*
 * Complete the 'solve' function below.
 *
 * The function is expected to return an INTEGER.
 * The function accepts INTEGER A as parameter.
 */

int solve(int A) {

}

> int main()
```