**2. Counting Closed Paths**

Some numbers are formed with closed paths. The digits *0, 4, 6* and *9* each have *1* closed path, and *8* has *2.*  None of the other numbers is formed with a closed path.Given a *number*, determine the total number of *closed paths* in all of its digits combined.

**Example**

*number = 649578*

The digits with closed paths are *6, 4, 9* and *8.*  The total number of closed paths is *1 + 1 + 1 + 2 = 5.*

**Function Description**

Complete the function *closedPaths* in the editor below.

closedPaths has the following parameter(s):

*int number:*  an integer

**Returns**:

*int:* the number of *closed paths* in *number*

**Constraints**

* 1 ≤*number*≤ 109

Input Format For Custom Testing

Input from stdin will be processed as follows and passed to the function:

The only line of input contains a single integer, *number*.

Sample Case 0

**Sample Input**

STDIN     Function

-----     ---------

630    →  number = 630

**Sample Output**

2

**Explanation**

Sum the *closed paths* count for each digit, *6, 3* and *0*. Return *1 + 0 + 1 = 2*.

Sample Case 1

**Sample Input**

STDIN     Function

-----     ---------

1288 →  number = 1288

**Sample Output**

4

**Explanation**

Sum the *closed paths* count for each digit, *1, 2, 8, 8*. Return *0 + 0 + 2 + 2 = 4*.

Language

C#



Autocomplete Ready

More

23

22

24

25

26

27

28

20

21

     \* The function is expected to return an INTEGER.

     \*

     \* The function accepts INTEGER number as parameter.

     \*/

static int[] hole={1,0,0,0,1,0,1,0,2,1};

    public static int closedPaths(int num)

    {

    /\*

     \* Complete the 'closedPaths' function below.



Line: 27 Col: 42

Test Results

Custom Input

Run Code

Run Tests

Submit

**Compiled successfully.All available test cases passed**

**Test case 0**

Test case 1

Test case 2

Test case 3

Test case 4

Test case 5

Test case 6

Test case 7

Test case 8

Test case 9

Test case 10

Input (stdin)

Run as Custom Input

|

Download

* **630**

Your Output (stdout)

* **2**

Expected Output

Download

* **2**

holes