PL-2022-C-Holes in a Number

locked

You are designing a poster which prints out numbers with a unique style applied to each of them. The styling is based on the number of closed paths or holes present in a given number.

The number of holes that each of the digits from 0 to 9 have are equal to the number of closed paths in the digit. Their values are:

1, 2, 3, 5, and 7 = 0 holes. 0, 4, 6, and 9 = 1 hole. 8 = 2 holes.

Given a number, you must determine the sum of the number of holes for all of its digits. For example, the number 819 has 3 holes.

Complete the program, it must must return an integer denoting the total number of holes in num.

Input Format

There is one line of text containing a single integer num, the value to process.

Constraints

1 ≤ num ≤ 10^9

Output Format

-

Sample Input 0

```
630
```

Sample Output 0

```
2
```

Explanation 0

Add the holes count for each digit, 6, 3 and 0. Return 1 + 0 + 1 = 2.

Sample Input 1

```
1288
```

Sample Output 1

```
4
```

Explanation 1

Add the holes count for each digit, 1, 2, 8, 8. Return 0 + 0 + 2 + 2 = 4.

Submissions: 759
Max Score: 100
Difficulty: Medium

Rate This Challenge:

More

C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>

int main() {

    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    int n,h,a;
    scanf("%d",&n);
    int hole[]={1,0,0,0,1,0,1,0,2,1};
    h=0;
    while(n>0){
        a=n%10;
        h+=hole[a];
        n/=10;
    }
    printf("%d",h);
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code