Status	Finished
Started	Friday, 31 October 2025, 12:50 PM
Completed	Friday, 31 October 2025, 1:25 PM
Duration	34 mins 58 secs

Question **1**

Correct

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.

Constraints

1 ≤ num ≤ 109

Input Format For Custom Testing

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

Sample Input

630

Sample Output

2

Explanation

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

Sample Case 1

Sample Input

1288

Sample Output

4

Explanation

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

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n,s=0,k;
    scanf("%d",&n);
    while(n!=0)
    {
        k=n%10;
        if(k==0||k==4||k==6||k==9)
        s+=1;
        else if(k==8)
        s+=2;
        else
        s+=0;
        n/=10;
    }
    printf("%d",s);
}
```

	Input	Expected	Got	
✓	630	2	2	✓
✓	1288	4	4	✓

Passed all tests! ✓

Question **2**

Correct

The problem solvers have found a new Island for coding and named it as Philaland. These smart people were given a task to make a purchase of items at the Island easier by distributing various coins with different values. Manish has come up with a solution that if we make coins category starting from $1 till the maximum price of the item present on Island, then we can purchase any item easily. He added the following example to prove his point.

Let's suppose the maximum price of an item is 5$ then we can make coins of {$1, $2, $3, $4, $5}to purchase any item ranging from $1 till $5.

Now Manisha, being a keen observer suggested that we could actually minimize the number of coins required and gave following distribution {$1, $2, $3}. According to him any item can be purchased one time ranging from $1 to $5. Everyone was impressed with both of them. Your task is to help Manisha come up with a minimum number of denominations for any arbitrary max price in Philaland.

Input Format

Contains an integer N denoting the maximum price of the item present on Philaland.

Output Format

Print a single line denoting the minimum number of denominations of coins required.

Constraints

1<=T<=100
1<=N<=5000

Refer the sample output for formatting

Sample Input 1:

10

Sample Output 1:

4

Sample Input 2:

5

Sample Output 2:

3

Explanation:

For test case 1, N=10.

According to Manish {$1, $2, $3,... $10} must be distributed.

But as per Manisha only {$1, $2, $3, $4} coins are enough to purchase any item ranging from $1 to $10. Hence minimum is 4. Likewise denominations could also be {$1, $2, $3, $5}. Hence answer is still 4.

For test case 2, N=5.

According to Manish {$1, $2, $3, $4, $5} must be distributed.

But as per Manisha only {$1, $2, $3} coins are enough to purchase any item ranging from $1 to $5. Hence minimum is 3. Likewise, denominations could also be {$1, $2, $4}. Hence answer is still 3.

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main()
{
    int n,c=0;
    scanf("%d",&n);
    while(n!=0)
     {    n/=2;
         c++;
}    printf("%d",c);
}
```

	Input	Expected	Got	
✓	10	4	4	✓
✓	5	3	3	✓
✓	20	5	5	✓
✓	500	9	9	✓
✓	1000	10	10	✓

Passed all tests! ✓

Question **3**

Correct

Problem Statement:

Integers are continuously entered by the user, one per line. The program must keep accepting integers until the user enters a negative number. The program must print each entered non-negative integer immediately as it is read.

Input Format:

Each line will contain one integer entered by the user.

Input terminates when a negative number is entered.

Boundary Conditions:

The number of integers entered can vary.

Each integer can range from -99999999 to 99999999.

Sample Input:

```
5
10
8
0
-3
```

Sample Output:

```
You entered: 5
You entered: 10
You entered: 8
You entered: 0
```

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int n;
    while(1){
        scanf("%d",&n);
        if(n<0)
        break;
        printf("You entered: %d\n",n);
    }
}
```

	Input	Expected	Got	
✓	5 10 8 0 -3	You entered: 5 You entered: 10 You entered: 8 You entered: 0	You entered: 5 You entered: 10 You entered: 8 You entered: 0	✓
✓	3 7 12 9 -1	You entered: 3 You entered: 7 You entered: 12 You entered: 9	You entered: 3 You entered: 7 You entered: 12 You entered: 9	✓

Passed all tests! ✓