Started on	Saturday, 30 December 2023, 7:13 AM
State	Finished
Completed on	Saturday, 30 December 2023, 8:13 AM
Time taken	59 mins 59 secs

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

$1 \leq num \leq 10^9$

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.

For example:

Input	Result
630	2
1288	4

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class Holes
{
    public static void main(String[] args)
    {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        int b[]=new int[100];
        int t=n;
        int rem=0,c=0,tot=0,ind=0;
        while(t!=0)
        {
            rem=t%10;
            c++;
            t/=10;
            b[ind++]=rem;
        }
        int a[]=new int[]{1,0,0,0,1,0,1,0,2,1};
        for(int i=0;i<c;i++)
```

```
20        {
21            for(int j=0;j<10;j++)
22            {
23                if(b[i]==j)
24                {
25                    tot+=a[j];
26                    break;
27                }
28            }
29        }
30        System.out.println(tot);
31    }
32 }
```

	Input	Expected	Got	
✔	630	2	2	✔
✔	1288	4	4	✔

Passed all tests! ✔

Alice and Bob are playing a game called "Stone Game". Stone game is a two-player game. Let N be the total number of stones. In each turn, a player can remove either one stone or four stones. The player who picks the last stone, wins. They follow the "Ladies First" norm. Hence Alice is always the one to make the first move. Your task is to find out whether Alice can win, if both play the game optimally.

Input Format

First line starts with T, which is the number of test cases. Each test case will contain N number of stones.

Output Format

Print "Yes" in the case Alice wins, else print "No".

Constraints

1<=T<=1000

1<=N<=10000

Sample Input and Output

Input

3

1

6

7

Output

Yes

Yes

No

For example:

Input	Result
3	Yes
1	Yes
6	No
7	

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class StoneGame
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int tc=sc.nextInt();
        for(int i=0;i<tc;i++)
        {
            int n=sc.nextInt();
            int t=n;
            int c=0;
            while(t!=0)
            {
                if((t-4)>=0)
                {
                    t-=4;
                    c++;
                }
                else{
                    t-=1;
                    c++;
                }
            }
            if(c%2==1)
            {
```

```
27                System.out.println("Yes");
28            }
29            else
30            {
31                System.out.println("No");
32            }
33        }
34    }
35 }
```

	Input	Expected	Got	
✔	3 1 6 7	Yes Yes No	Yes Yes No	✔

Passed all tests! ✔

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.

Input Format:

The first line will contain the N numbers separated by one space.

Boundary Conditions:

3 <= N <= 50

The value of the numbers can be from -99999999 to 99999999

Output Format:

The count of numbers where the numbers are odd numbers.

Example Input / Output 1:

Input:

5 10 15 20 25 30 35 40 45 50

Output:

5

Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

For example:

Input	Result
5 10 15 20 25 30 35 40 45 50	5

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class Odd
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String n = sc.nextLine();

        int size=n.length();
        int s=(size/2)+(size%2);

        int a[]=new int[size];
        for(int i=0;i<size;i++)
        {
            char ch=n.charAt(i);

            if(ch!=' ')
            {
                int val=0;
                val=(val*10)+n.charAt(i);

            }
            else
            {
            a[ind]=val;
            }
        }
        System.out.println();
        // int ind=0;
        // int a[]=new int[s];
        // for(int i=0;i<size;i++)
        // {
        //
        //      if()
        //      {
        //
```

```
37        //      }
38        //      else
39        //      {
40        //
41        //      }
42        //  }
43        //  for(int i=0;i<size;i++)
44        //  {
45        //      System.out.print(a[i]+" ");
46        //  }
47      }
48  }
```

Syntax Error(s)

```
Odd.java:25: error: cannot find symbol
         a[ind]=val;
          ^
 symbol:   variable ind
 location: class Odd
Odd.java:25: error: cannot find symbol
         a[ind]=val;
                ^
 symbol:   variable val
 location: class Odd
2 errors
```

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.

For example:

Input	Result
10	4
5	3
20	5

Answer: (penalty regime: 0 %)

1	