PREETHI .B

ECE-D

240801251

Problem Statement:

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

3

1

6

7

Sample Output

Yes

Yes

No

Answer: (penalty regime: 0 %)

```
#include<stdio.h>
int main(){
    int T,N;
    scanf("%d",&T);
    while(T--){
        scanf("%d",&N);
        int alice=1;
    while(N>0){
        if(alice){
            if (N>=4){
                N-=4;
            }
            else{
                N-=1;
            }
        }
        else{
            if(N>=4){
                N-=4;
            }
            else{
                N-=1;
            }
        }
        alice=!alice;
    }
        if(alice){
            printf("No\n");
        }
        else{
            printf("Yes\n");
        }

    }
    return 0;
}
```

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

Passed all tests! ✓

Problem Statement:

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, 7 = 0 holes.

0, 4, 6, 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 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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Problem Statement:

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

Sample Input 1:

10

Sample Output 1:

4

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    int a,c;
    scanf("%d",&a);
    int d=1;
    while(d<=a){
            c++;
            d*=2;
        }

    printf("%d",c);

}
```

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

Passed all tests! ✓

Problem Statement:

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.

Sample Input:

5 10 15 20 25 30 35 40 45 50

Sample Output:

5

Answer: (penalty regime: 0 %)

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

	Input	Expected	Got	
✓	5 10 15 20 25 30 35 40 45 50	5	5	✓

Passed all tests! ✓

Problem Statement:

Given a number N, return true if and only if it is a confusing number, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180

degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180

degrees, they become invalid. A confusing number is a number that when rotated 180

degrees becomes a different number with each digit valid.

Example 1:

Input: 6

Output: true

Explanation: We get 9 after rotating 6, 9 is a valid number and 9 != 6.

Example 2:

Input: 89

Output: true

Explanation: We get 68 after rotating 89, 86 is a valid number and 86 != 89.

Example 3:

Input: 11

Output: false

Explanation: We get 11 after rotating 11, 11 is a valid number but the value remains the

same, thus 11 is not a confusing number.

Example 4:

Input: 25

Output: false

Explanation: We get an invalid number after rotating 25.

Note:

1. 0 <= N <= 10^9

2. After the rotation we can ignore leading zeros, for example if after rotation we have

0008 then this number is considered as just 8.

Answer: (penalty regime: 0 %)

```
#include <stdio.h>

int main(){
    int a,b,c;
    scanf("%d",&a);
    while(a>0){
        b=a%10;
        if(b==2||b==3||b==4||b==5||b==7){
            c+=1;
        }
        a=a/10;
    }
    if(c>0){
        printf("false");
    }
    else{
        printf("true");
    }
}
```

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

Problem Statement:

A nutritionist is labeling all the best power foods in the market. Every food item

arranged in a single line, will have a value beginning from 1 and increasing by 1 for each,

until all items have a value associated with them. An item's value is the same as the number

of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food

item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum

total of macronutrients. However, the nutritionist must avoid prescribing a particular sum

of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses

food items in the increasing order of their value. Compute the highest total of

macronutrients that can be prescribed to a patient, without the sum matching the given

'unhealthy' number.

Here's an illustration: Given 4 food items (hence value: 1,2,3 and 4), and the

unhealthy sum being 6 macronutrients, on choosing items 1, 2, 3 -> the sum is 6, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best

combination is from among:

• 2 + 3 + 4 = 9

• 1 + 3 + 4 = 8

• 1 + 2 + 4 = 7

Since 2 + 3 + 4 = 9, allows for maximum number of macronutrients, 9 is the right

answer. Complete the code in the editor below. It must return an integer that represents

the maximum total of macronutrients, modulo 1000000007 (109 + 7).

It has the following:

n: an integer that denotes the number of food items

k: an integer that denotes the unhealthy number

Constraints

• 1 ≤ n ≤ 2 × 109

• 1 ≤ k ≤ 4 × 1015

Input Format For Custom Testing

The first line contains an integer, n, that denotes the number of food items. The second line

contains an integer, k, that denotes the unhealthy number.

Sample Input 0

2

2

Sample Output 0

3

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main(){
    long n,k,i,sum;
    scanf("%ld %ld",&n,&k);
    for(i=0;i<=n;i++){
        sum=sum+i;
        if(sum==k){
            sum=sum-1;
        }
    }
    printf("%ld",sum%1000000007);
}
```

	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓