Narmadha

S

ECE-D

240801216

ProblemStatement:

AliceandBobareplayingagamecalled"StoneGame".Stonegameisa two-playergame.

LetNbethetotalnumberofstones.Ineachturn,aplayercanremoveeither onestoneor

fourstones.Theplayerwhopicksthelaststone,wins.Theyfollowthe"Ladies First"norm.

HenceAliceisalwaystheonetomakethefirstmove.Yourtaskistofindout whetherAlice

canwin,ifbothplaythegameoptimally.

InputFormat

FirstlinestartswithT,whichisthenumberoftestcases.Eachtestcasewill containN

numberofstones.

OutputFormat

Print"Yes"inthecaseAlicewins,elseprint"No".

Constraints1<=T<=10001<=N<=10000
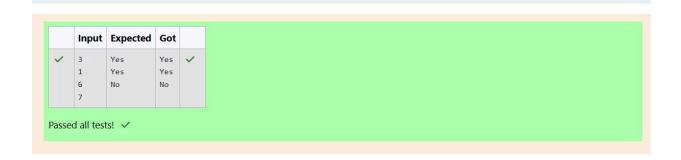
SampleInput

3

1

6

7

SampleOutput

Yes

Ye

s

No

Answer:(penaltyregime:0%)

```c
1    #include<stdio.h>
2    intmain(){
3        1ntT,N;
4        scant("id",IT);
5        while(T--)(
6            scanf("°d",&N);
7            int  a1ice=1;
8        while(NGO)(
9            if(alice){
10               if(N>=4){
11                   N-=4;
12               }
13               else{
14                   N-=1;
15               }
16           }
17           edse(
18               if(N>=4){
19                   N-=4;
20               }
21               else{
22                   N-=1;
23               }
24           }
25           al1ce=!a11ce;
26       }
27       if(alice)(
28           pristf("No',re");

30       else{
31           printf("Yes\n");
32       }
33

35       return 0;
```

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

Passed all tests! ✓

ProblemStatement:

Youaredesigningaposterwhichprintsoutnumberswithauniquestyle appliedtoeach

ofthem.Thestylingisbasedonthenumberofclosedpathsorholespresent inagiven

number.

Thenumberofholesthateachofthedigitsfrom0to9haveareequaltothe numberof

closedpathsinthedigit.Theirvaluesare:

1,2,3,5,7=0holes.

0,4,6,9=1hole.

8=2holes.

Givenanumber,youmustdeterminethesumofthenumberofholesforallof itsdigits.

Forexample,thenumber819has3holes.

Complete the program, it 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

```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! ✓

ProblemStatement:

TheproblemsolvershavefoundanewIslandforcodingandnameditas Philaland.These

smartpeopleweregivenatasktomakeapurchaseofitemsattheIsland easierby

distributingvariouscoinswithdifferentvalues.Manishhascomeupwitha solutionthatif

wemakecoinscategorystartingfrom$1tillthemaximumpriceoftheitem presenton

Island,thenwecanpurchaseanyitemeasily.Headdedthefollowingexample toprove

hispoint.

Let'ssupposethemaximumpriceofanitemis5$thenwecanmakecoinsof {$1,$2,$3,

$4,$5}topurchaseanyitemrangingfrom$1till$5.

NowManisha,beingakeenobserversuggestedthatwecouldactually minimizethe

numberofcoinsrequiredandgavefollowingdistribution{$1,$2,$3}. Accordingtohim

anyitemcanbepurchasedonetimerangingfrom$1to$5.Everyonewas impressedwith

bothofthem.YourtaskistohelpManishacomeupwithaminimumnumber

of

denominations for any arbitrary max price in Philaland.

InputFormat

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

SampleInput1:

10

SampleOutput1:

4

```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! ✓

ProblemStatement:

AsetofNnumbers(separatedbyonespace)ispassedasinputtothe program.The

programmustidentifythecountofnumberswherethenumberisodd number.

InputFormat:

ThefirstlinewillcontaintheNnumbersseparatedbyonespace.

BoundaryConditions:

3<=N<=50

Thevalueofthenumberscanbefrom-99999999to99999999

OutputFormat:

Thecountofnumberswherethenumbersareoddnumbers.

SampleInput:

5101520253035404550

SampleOutput:

5

Answer: (penalty regime: 0 %)

```c
#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! ✓

ProblemStatement:

GivenanumberN,returntrueifandonlyifitisaconfusingnumber,which
satisfiesthe

followingcondition:

Wecanrotatedigitsby180degreestoformnewdigits.When0,1,6,8,9are rotated180

degrees,theybecome0,1,9,8,6respectively.When2,3,4,5and7are rotated180

degrees,theybecomeinvalid.Aconfusingnumberisanumberthatwhen rotated180

degreesbecomesadifferentnumberwitheachdigitvalid.

Example1:

Input:6

Output:true

Explanation:Weget9afterrotating6,9isavalidnumberand9!=6.

Example2:

Input:89

Output:true

Explanation:Weget68afterrotating89,86isavalidnumberand86!=89.

Example3:

Input:11

Output:false

Explanation:Weget11afterrotating11,11isavalidnumberbutthevalue remainsthe

same,thus11isnotaconfusingnumber.

Example4:

Input:25

Output:false

Explanation:Wegetaninvalidnumberafterrotating25. Note:

1. 0<=N<=10^9

2. Aftertherotationwecanignoreleadingzeros,forexampleifafterrotation wehave

0008thenthisnumberisconsideredasjust8.

```c
#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 (10^9+7). It

has the following:

n: an integer that denotes the number of food items

k: an integer that denotes the unhealthy number

Constraints

- $1 \le n \le 2 \times 10^9$

- $1 \le k \le 4 \times 10^{15}$

InputFormatForCustomTesting

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.

SampleInput0

2

2

SampleOutput0

3

Answer: (penalty regime: 0 %)

```c
 1  #include <stdio.h>
 2  int main(){
 3      long n,k,i,sum;
 4      scanf("%ld %ld",&n,&k);
 5      for(i=0;i<=n;i++){
 6          sum=sum+i;
 7          if(sum==k){
 8              sum=sum-1;
 9          }
10      }
11      printf("%ld",sum%1000000007);
12  }
```

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

Passed all tests! ✓