paranidharan.R

ECE-D

240801238

ProblemStatement1:

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".

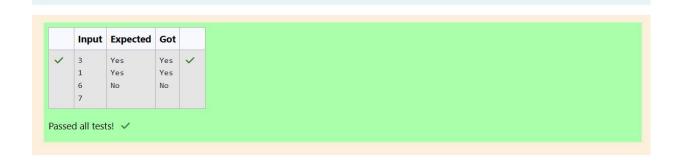Constraints1<=T<=10001<=N<=10000

SampleInput

3

1

6

7

SampleOutput

Yes

Yes

No

Answer: (penalty regime: 0 %)

```c
#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! ✓

ProblemStatement2:

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

$1 \leq num \leq 10^9$

InputFormatForCustomTesting

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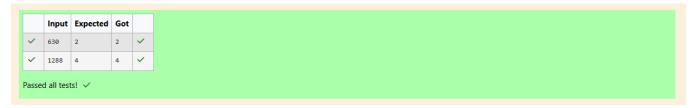
SampleInput

630

SampleOutput

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! ✓

ProblemStatement3:

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

denominationsforanyarbitrarymaxpriceinPhilaland.

InputFormat

ContainsanintegerNdenotingthemaximumpriceoftheitempresenton Philaland.

Output Format

Printasinglelinedenotingtheminimumnumberofdenominationsofcoins required.

Constraints

1<=T<=1001<=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! ✓
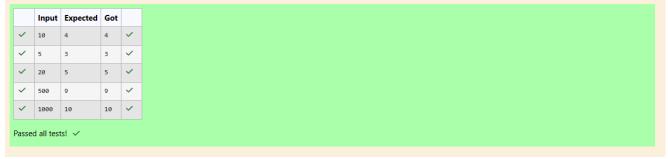
ProblemStatement4:

AsetofNnumbers(separatedbyonespace)ispassedasinputtothe program.The

programmustidentifythecountofnumberswherethenumberisodd number.

InputFormat:

ThefirstlinewillcontaintheNnumbersseparatedbyonespace.

BoundaryConditions:

3<=N<=50

Thevalueofthenumberscanbefrom-99999999to99999999

OutputFormat:

Thecountofnumberswherethenumbersareoddnumbers.

SampleInput:

5101520253035404550

SampleOutput:

5

```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! ✓

ProblemStatement5:

GivenanumberN,returntrueifandonlyifitisaconfusingnumber,which satisfiesthe

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.

Example1:

Input: 6

Output: true

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

Example2:

Input: 89

Output: true

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

Answer: (penalty regime: 0 %)

```c
1  #include <stdio.h>
2
3  int main(){
4      int a,b,c;
5      scanf("%d",&a);
6      while(a>0){
7          b=a%10;
8          if(b==2||b==3||b==4||b==5||b==7){
9              c+=1;
10         }
11         a=a/10;
12     }
13     if(c>0){
14         printf("false");
15     }
16     else{
17         printf("true");
18     }
19 }
```

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

Passed all tests! ✓

ProblemStatement6:

Anutritionistislabelingallthebestpowerfoodsinthemarket.Everyfood item

arrangedinasingleline,willhaveavaluebeginningfrom1andincreasingby 1foreach,

untilallitemshaveavalueassociatedwiththem.Anitem'svalueisthesame asthenumber

ofmacronutrientsithas.Forexample,fooditemwithvalue1has1 macronutrient,food

itemwithvalue2has2macronutrients,andincrementinginthisfashion.

Thenutritionisthastorecommendthebestcombinationtopatients,i.e. maximum

totalofmacronutrients.However,thenutritionistmustavoidprescribinga particularsum

ofmacronutrients(an'unhealthy'number),andthissumisknown.The nutritionistchooses

fooditemsintheincreasingorderoftheirvalue.Computethehighesttotalof

macronutrientsthatcanbeprescribedtoapatient,withoutthesum matchingthegiven

'unhealthy'number.

Here'sanillustration:Given4fooditems(hencevalue:1,2,3and4),andthe

unhealthysumbeing6macronutrients,onchoosingitems1,2,3->thesumis 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
#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! ✓