WEEK 4

StoneGame-OneFour

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 fourstones.Theplayerwhopicksthelaststone,wins.Theyfollowthe"LadiesFirst"norm. HenceAliceisalwaystheonetomakethefirstmove.YourtaskistofindoutwhetherAlice can win, if both play the game optimally.

Input Format

FirstlinestartswithT,whichisthenumberoftestcases.EachtestcasewillcontainN number of stones.

Output Format

Print"Yes"inthecaseAlicewins,elseprint"No".

Constraints 1<=T<=1000 1<=N<=10000

SampleInput

3

1

6

7

SampleOutput

Yes

Yes

No

```c
#include<stdio.h>
int main()
{
    int T,N,rem,mod;
    scanf("%d", &T);
    while (T--)
    {
        scanf("%d",&N);
        if(N%4 == 0)
        {
            rem = N/4;
            if(rem% 2 != 0)
            {
                printf("Yes\n");

            }
            else
            {
                printf("No\n");

            }

        }
        else
        {
            rem = N/4;
            mod = N%4;
            if(rem% 2!= 0 )
            {
                if (mod == 1 || mod == 3)
                {
                    printf("No\n");

                }
                else
                {
                    printf("Yes\n");

                }

            }
            else
            {
                if(mod == 1 || mod == 3)
                printf("Yes\n");
                else
                printf("No\n");
            }
        }
    }
    return 0;
}
```

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

Passed all tests! ✓

Holesin a Number

Problem Statement

ofthem.Thestylingisbasedonthenumberofclosedpathsorholespresentinagiven number.

Thenumberofholesthateachofthedigitsfrom0to9haveareequaltothenumberof closed paths in the digit. Their values are:

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

0, 4, 6, 9 =1 hole.

8 =2 holes.

Givenanumber,youmustdeterminethesumofthenumberofholesforallofitsdigits. For example, the number 819 has 3 holes.

Completetheprogram,itmustreturnanintegerdenotingthetotalnumberofholesin num.

Constraints

1 ≤num ≤109

InputFormatForCustomTesting

Thereisonelineoftextcontainingasingleintegernum,thevaluetoprocess. Sample Input

630

SampleOutput

2

```c
#include<stdio.h>
int countHoles(int num)
{
    int holes = 0;
    while (num > 0)
    {
        int digit = num % 10;
        switch (digit)
        {
            case 0:
            case 4:
            case 6:
            case 9:
                holes += 1;
                break;
            case 8:
                holes += 2;
                break;
        }
        num /= 10;
    }
    return holes;
}
int main()
{
    int num;
    scanf("%d", &num);
    int result = countHoles(num);
    printf("%d\n", result);
    return 0;
}
```

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

Passed all tests! ✓

PhilalandCoin

Problem Statement:

Theproblem 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 distributingvariouscoinswithdifferentvalues.Manishhascomeupwithasolutionthatif 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 provehis point.

Let'ssupposethemaximumpriceofanitemis5$ thenwecanmakecoinsof{$1,$2,$3, $4,$5}topurchase anyitemrangingfrom$1till $5.

Now Manisha, being a keen observer suggested that we could actually minimize the numberof coins required and gave following distribution {$1, $2, $3}.According to him anyitemcanbepurchasedonetimerangingfrom$1to$5.Everyonewasimpressedwith 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

ContainsanintegerNdenotingthemaximumpriceoftheitempresentonPhilaland. Output Format

Printasinglelinedenotingtheminimumnumberofdenominationsofcoinsrequired. Constraints

1<=T<=1001<=N<=5000

SampleInput1:

10

SampleOutput 1:

4

```c
#include<stdio.h>
int minDenominations(int N)
{
    int count = 0;
    int sum = 0;
    int denomination = 1;
    while (sum < N)
    {
        sum += denomination;
        denomination *= 2;
        count++;
    }
    return count;
}
int main()
{
    int N;
    scanf("%d", &N);
    int result = minDenominations(N);
    printf("%d\n",result);
    return 0;
}
```

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

Passed all tests! ✓

NumberCount

Problem Statement:

AsetofNnumbers(separatedbyonespace)ispassedasinputtotheprogram.The program must identify the count of numbers where the number is odd number.

Input Format:

ThefirstlinewillcontaintheNnumbersseparatedbyonespace. Boundary Conditions:

3<=N<=50

Thevalueofthenumberscanbefrom-99999999to99999999 Output Format:

Thecount ofnumberswherethe numbersareoddnumbers.

Sample Input:

5 10 15 20 25 30 35 40 45 50

SampleOutput:

5

```c
#include<stdio.h>
int main()
{
    int N=10,i=0,a,sum=0;
    while(i<N)
    {
        scanf("%d", &a);
        if(a%2 != 0)
        {
            sum += 1;
        }
        i++;
    }
    printf("%d",sum);
    return 0;
}
```

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

Passed all tests! ✓

Confusing Number

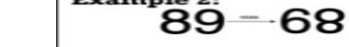
Problem Statement:

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.

```c
#include<stdio.h>
#include<stdbool.h>
bool isConfusingNumber(int N)
{
    int rotatedDigits[]={0,1,-1,-1,-1,-1,9,-1,8,6};
    int original = N;
    int rotatedNumber = 0;
    int placeValue = 1;
    while(N>0)
    {
        int digit = N%10;
        if(rotatedDigits[digit]==-1)
        {
            return false;
        }
        rotatedNumber += rotatedDigits[digit]*placeValue;
        placeValue *= 10;
        N/=10;
    }
    return rotatedNumber != original;
}
int main()
{
    int N;
    scanf("%d", &N);
    if(isConfusingNumber(N))
    {
        printf("true\n");
    }
    else
    {
        printf("false\n");
    }
    return 0;
}
```

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

Passed all tests! ✓

NutritionValue

Problem Statement:

Anutritionistislabelingallthebestpowerfoodsinthemarket.Everyfooditem arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, untilallitemshaveavalueassociatedwiththem.Anitem'svalueisthesameasthenumber 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.

Thenutritionisthasto recommendthebestcombination topatients,i.e. maximum

total of macronutrients. However, the nutritionist must avoid prescribing a particular sum ofmacronutrients(an'unhealthy'number),andthissumisknown.Thenutritionistchooses 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 anillustration:Given4fooditems(hence value:1,2,3and4),andthe unhealthy sum being 6 macronutrients, on choosing items1, 2, 3 ->the sum is6, which matchesthe'unhealthy'sum.Hence,oneofthethreeneedstobeskipped.Thus,thebest 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.Completethecodeintheeditorbelow.Itmustreturnanintegerthatrepresents the maximum total of macronutrients, modulo 1000000007 (109 + 7).

Ithasthe following:

n:anintegerthatdenotesthenumberoffooditems k: an integer that denotes the unhealthy number

Constraints

• $1 \leq n \leq 2 \times 109$

• $1 \leq k \leq 4 \times 1015$

InputFormatForCustomTesting

Thefirstlinecontainsaninteger,n,thatdenotesthenumberoffooditems.Thesecondline contains an integer, k, that denotes the unhealthy number.

SampleInput0

2

2

SampleOutput 0

3

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

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