**Count rotations divisible by 4**

Show Topic Tags   

Given a large positive number N having x digits, the task is to count all rotations of the given number which are divisible by 4.

Examples:

Input: 8

Output: 1

Input: 20

Output: 1

Rotation: 20 is divisible by 4

02 is not divisible by 4

**Input:**  
The first line of input contains an integer T denoting the no of test cases. Then T test cases follow. Each test case contains a number N.  
  
**Output:**  
For each test case in a new line print the count of all rotations of N which are divisible by 4.  
  
**Constraints:**  
1<=T<=100  
1<=x<=10^5  
  
**Example:  
Input:**  
2  
8  
20  
**Output:**  
1  
1

\*\*For More Examples Use Expected Output\*\*

<http://practice.geeksforgeeks.org/problems/count-rotations-divisible-by-4/0>

import java.util.\*;

import java.lang.\*;

import java.io.\*;

class GFG {

static char[] RotarDerecha(char[] arr, int k)

{

k = k % arr.length;

char[] rotado = new char[arr.length];

int i = 0;

for (i = k; i < rotado.length; i++)

{

rotado[i] = arr[i - k];

}

for (int j = i - k; j < arr.length; j++)

{

rotado[j - (i - k)] = arr[j];

}

return rotado;

}

static String concat(int[] a)

{

String con = "";

for (int i = 0; i < a.length; i++)

{

con += (a[i]);

}

return con;

}

public static void main(String[] args) throws IOException {

// TODO code application logic here

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

int t = Integer.parseInt(br.readLine());

while(t-- > 0) {

//char[] s = br.readLine().trim().toCharArray();

String s = br.readLine().trim();

int ans = 0;

if (s.length() < 3)

{

if (Integer.parseInt(s) % 4 == 0)

{

ans++;

}

}

else

{

if (Integer.parseInt(s.substring(s.length() - 2)) % 4 == 0)

{

ans++;

}

char[] rot = s.toCharArray();

for (int i = 0; i < s.length()-1; i++)

{

rot = RotarDerecha(rot, 1);

if (Integer.parseInt(new String(rot).substring(rot.length - 2)) % 4 == 0)

{

ans++;

}

}

}

System.out.println(ans);

}

}

}