**No of Carry Operations**

Show Topic Tags   

Given two numbers, the task is to find the number of carry operations required when two numbers are added as below..  
1234  
+  
5678  
--------  
6912  
--------  
4+8 = 2 and carry 1  
carry+3+7 = carry 1  
carry+2+6 = 9, carry 0  
carry+1+5 = 6

We need two carry operations in this example.

**Input:**  
First line of input consists of an integer T, denoting the no of test cases. Then T test cases follow. Each test case contains two unsigned integers(n1,n2) less than 10 digits.  
  
**Output:**  
For each test case in a new line print the number of carry operations that would be required while adding n1 and n2.  
  
**Constraints:**  
1<=T<=50  
1<=n1, n2 <= 9\*10^10  
  
**Example:**  
**Input:**  
3  
123 456  
555 555  
123 594

**Output:**  
0  
3  
1

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

<http://practice.geeksforgeeks.org/problems/no-of-carry-operations/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication243;

import java.io.\*;

import java.util.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication243 {

/\*\*

\* @param args the command line arguments

\*/

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) {

String[] input = br.readLine().trim().split(" ");

//2345789456

long a = Long.parseLong(input[0]);

long b = Long.parseLong(input[1]);

int ans = 0;

int carry = 0;

while (a > 0 && b > 0)

{

long sum = a % 10 + b % 10 + carry;

if (sum > 9)

{

carry = 1;

ans++;

}

else

{

carry = 0;

}

a /= 10;

b /= 10;

}

while (a > 0)

{

long sum = a % 10 + carry;

if (sum > 9)

{

ans++;

carry = 1;

}

else

{

carry = 0;

}

a /= 10;

}

while (b > 0)

{

long sum = b % 10 + carry;

if (sum > 9)

{

ans++;

carry = 1;

}

else

{

carry = 0;

}

b /= 10;

}

System.out.println(ans);

}

}

}