**All numbers with specific difference**

Show Topic Tags   

Given a positive number N and a number D. Write a program to find the count of numbers smaller than N such that the difference between the number and sum of its digits is greater than or equal to given specific value D.

**Input:**  
First line of input contains a single integer T which denotes the number of test cases. Then T test cases follows. First line of each test case contains two space separated integers N and D.  
**Output:**  
For each test case, print the count of numbers smaller than N such that the difference between the number and sum of its digits is greater than or equal to D.

**Constraints:**  
1<=T<=100  
1<=N<=105  
1<=D<=104

**Example:  
Input:**  
2  
13 2  
14 3  
**Output:**  
4  
5

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

Contributor: Harsh Agarwal

<http://practice.geeksforgeeks.org/problems/all-numbers-with-specific-difference/0>

/\*

\* To change this template, choose Tools | Templates

\* and open the template in the editor.

\*/

package javaapplication252;

import java.util.\*;

import java.lang.\*;

import java.io.\*;

/\*\*

\*

\* @author Administrador

\*/

public class JavaApplication252 {

/\*\*

\* @param args the command line arguments

\*/

static int sumDig(int n) {

String s = String.valueOf(n);

int sum =0;

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

sum += Integer.parseInt(String.valueOf(s.charAt(i)));

}

return sum;

}

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(" ");

int n = Integer.parseInt(input[0]);

int d = Integer.parseInt(input[1]);

int ans =0;

for(int i =1; i<n; i++) {

if(i-sumDig(i) >= d) {

ans++;

}

}

System.out.println(ans);

}

}

}

--------editorial-------

We can solve this problem by observing a fact that for a number k less than N,

if k – sumofdigit(k) >= diff then

above equation will be true for (k+1)

also because we know that sumofdigit(k+1)

is not greater than sumofdigit(k) + 1

so, k + 1 - sumofdigit(k + 1) >=

k - sumofdigit(k)

but we know that right side of above

inequality is greater than diff,

so left side will also be greater than

diff.

So, finally we can say that if a number k satisfies the difference condition then (k + 1) will also satisfy same equation so our job is to find the smallest number which satisfies the difference condition then all numbers greater than this and up to N will satisfy the condition so our answer will be N – smallest number we found.  
We can find the smallest number satisfying this condition using binary search so total time complexity of solution will be O(log N)

* C/C++
* Java

|  |
| --- |
| /\*  C++ program to count total numbers which     have difference with sum of digits greater     than specific value \*/  #include <bits/stdc++.h>  using namespace std;    //  Utility method to get sum of digits of K  int sumOfDigit(int K)  {      //  loop until K is not zero      int sod = 0;      while (K)      {          sod += K % 10;          K /= 10;      }      return sod;  }    // method returns count of numbers smaller than N,  // satisfying difference condition  int totalNumbersWithSpecificDifference(int N, int diff)  {      int low = 1, high = N;        //  binary search while loop      while (low <= high)      {          int mid = (low + high) / 2;            /\* if difference between number and its sum             of digit is smaller than given difference             then  smallest number will be on left side \*/          if (mid - sumOfDigit(mid) < diff)              low = mid + 1;            /\* if difference between number and its sum             of digit is greater than or equal to given             difference then  smallest number will be on             right side \*/          else              high = mid - 1;      }        // return the difference between 'smallest number      // found' and 'N' as result      return (N - high);  }    //  Driver code to test above methods  int main()  {      int N = 13;      int diff = 2;        cout << totalNumbersWithSpecificDifference(N, diff);      return 0;  } |

Run on IDE

Output:

4

This article is contributed by [**Utkarsh Trivedi**](https://in.linkedin.com/in/utkarsh-trivedi-253069a7). If you like GeeksforGeeks and would like to contribute, you can also write an article using [contribute.geeksforgeeks.org](http://www.contribute.geeksforgeeks.org/)or mail your article to contribute@geeksforgeeks.org. See your article appearing on the GeeksforGeeks main page and help other Geeks.

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above.