

code jam

printf("hello, world!\n");

Qualification Round 2015

[A. Standing Ovation](#)[B. Infinite House of Pancakes](#)**C. Dijkstra**[D. Ominous Omino](#)[Contest Analysis](#)[Ask a question](#)[View my submissions](#)

## - Submissions

## Standing Ovation

7pt	Not attempted 22964/26528 users correct (87%)
10pt	Not attempted 19346/22732 users correct (85%)

## Infinite House of Pancakes

9pt	Not attempted 7805/17231 users correct (45%)
12pt	Not attempted 5442/6704 users correct (81%)

## Dijkstra

11pt	Not attempted 6663/9721 users correct (69%)
17pt	Not attempted 2492/4819 users correct (52%)

## Ominous Omino

8pt	1 incorrect attempt 7342/9200 users correct (80%)
26pt	Not attempted 686/4030 users correct (17%)

## - Top Scores

kyc	100
ksun48	100
darnley	100
AntiForest	100
shik	100
Nicolas16	100

Practice Mode Rank: 23297 Score: 0varun.narisetty@gmail.com | [Contest scoreboard](#) | [Sign out](#)

## Problem C. Dijkstra

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input  
11 points

Solve C-small

Large input  
17 points

Solve C-large

## Problem

The Dutch computer scientist Edsger Dijkstra made many important contributions to the field, including the shortest path finding algorithm that bears his name. This problem is not about that algorithm.

You were marked down one point on an algorithms exam for misspelling "Dijkstra" -- between D and stra, you wrote some number of characters, each of which was either i, j, or k. You are prepared to argue to get your point back using *quaternions*, an actual number system (extended from complex numbers) with the following multiplicative structure:

	1	i	j	k
1	1	i	j	k
i	i	-1	k	-j
j	j	-k	-1	i
k	k	j	-i	-1

To multiply one quaternion by another, look at the row for the first quaternion and the column for the second quaternion. For example, to multiply *i* by *j*, look in the row for *i* and the column for *j* to find that the answer is *k*. To multiply *j* by *i*, look in the row for *j* and the column for *i* to find that the answer is *-k*.

As you can see from the above examples, the quaternions are not commutative -- that is, there are some *a* and *b* for which  $a * b \neq b * a$ . However they are associative -- for any *a*, *b*, and *c*, it's true that  $a * (b * c) = (a * b) * c$ .

Negative signs before quaternions work as they normally do -- for any quaternions *a* and *b*, it's true that  $-a * -b = a * b$ , and  $-a * b = a * -b = -(a * b)$ .

You want to argue that your misspelling was equivalent to the correct spelling *ijk* by showing that you can split your string of *is*, *js*, and *ks* in two places, forming three substrings, such that the leftmost substring reduces (under quaternion multiplication) to *i*, the middle substring reduces to *j*, and the right substring reduces to *k*. (For example, *jij* would be interpreted as  $j * i * j$ ;  $j * i$  is *-k*, and  $-k * j$  is *i*, so *jij* reduces to *i*.) If this is possible, you will get your point back. Can you find a way to do it?

ProjectYoung	100
azariamuh	100
wo...	100
ctunoku	100

## Input

The first line of the input gives the number of test cases,  $T$ .  $T$  test cases follow. Each consists of one line with two space-separated integers  $L$  and  $X$ , followed by another line with  $L$  characters, all of which are  $i$ ,  $j$ , or  $k$ . Note that the string never contains negative signs,  $1$ s, or any other characters. The string that you are to evaluate is the given string of  $L$  characters repeated  $X$  times. For instance, for  $L = 4$ ,  $X = 3$ , and the given string  $kijj$ , your input string would be  $kijjkijjkijj$ .

## Output

For each test case, output one line containing "Case #x: y", where  $x$  is the test case number (starting from 1) and  $y$  is either YES or NO, depending on whether the string can be broken into three parts that reduce to  $i$ ,  $j$ , and  $k$ , in that order, as described above.

## Limits

$1 \leq T \leq 100$ .  
 $1 \leq L \leq 10000$ .

## Small dataset

$1 \leq X \leq 10000$ .  
 $1 \leq L * X \leq 10000$ .

## Large dataset

$1 \leq X \leq 10^{12}$ .  
 $1 \leq L * X \leq 10^{16}$ .

## Sample

Input	Output
5	Case #1: NO
2 1	Case #2: YES
ik	Case #3: NO
3 1	Case #4: YES
ijk	Case #5: NO
3 1	
kji	
2 6	
ji	
1 10000	
i	

In Case #1, the string is too short to be split into three substrings.

In Case #2, just split the string into  $i$ ,  $j$ , and  $k$ .

In Case #3, the only way to split the string into three parts is  $k$ ,  $j$ ,  $i$ , and this does not satisfy the conditions.

In Case #4, the string is  $jijijijijiji$ . It can be split into  $jij$  (which reduces to  $i$ ),  $iji$  (which reduces to  $j$ ), and  $jijiji$  (which reduces to  $k$ ).

In Case #5, no matter how you choose your substrings, none of them can ever reduce to  $a$  or a  $k$ .

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2015 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

