# Keyboard

After the final of the last season of CodeIT Ivancho was left broke – he needed to buy a lot of cakes (let's not blame Encho's solution for that). In order to fix what's done, he decided to start a job.

His current task is to type a given text on a computer. In this case, it's a string with length N, which consists of lowercase letters of the English alphabet. Unfortunately, he types on a keyboard with only two fingers and it's all very slow. Ivancho's purpose is to complete the task as fast as possible i.e. to minimize the moves of his fingers, which he needs to make while typing the text. That's why he decided to rearrange the keys on his keyboard. Your task is to help him by suggesting a way to do this.

In order to evaluate your solution, the following result is assigned to it – the distance passed by Ivancho's fingers while moving though the keyboard. In the beginning, you tell Ivancho where to place his fingers (on which keys). For each letter of the text he moves one of his fingers from its previous position to the position of the current letter. In order to figure out which finger to use, the Euclidean distance is calculated (between the points in the center of each key) and the closest finger is chosen, in the case of equal distances we always use the left finger. It is not allowed to "entangle the fingers" i.e. to place the left finger on the right side of the right finger (nor is it allowed to place the two fingers on the same coordinates on Ox). In these situations the other finger is used. The passed distance from the current position of the chosen finger to the key of the current letter, rounded to an integer, is added to the result. If a finger is already placed on the key of the current letter, the result does not increase. Solutions which receive a lower result are evaluated as better.

For the calculation of the distance, the following standard keyboard blueprint will be used:
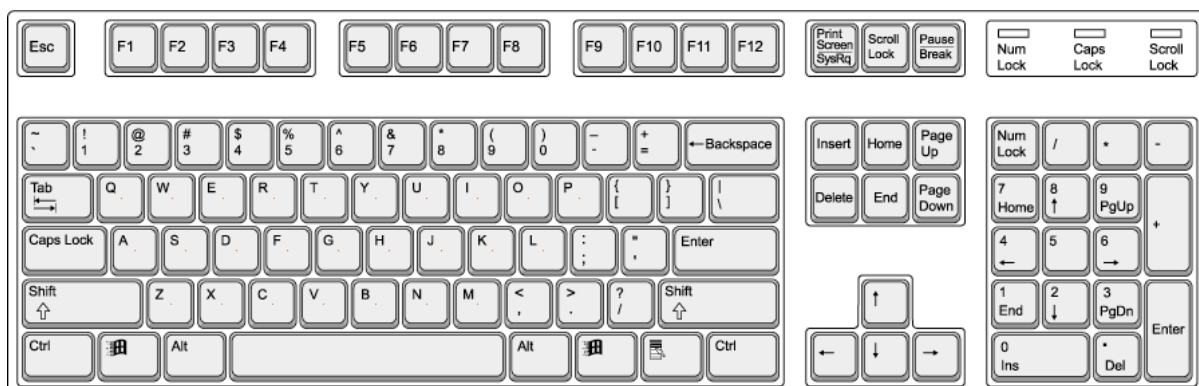
The central pixels which will be used for the calculations are colored in orange.

Let's number the keys from left to right and from top to bottom i.e. in the original arrangement of the keys the letter **q** is on key number 1, **w** is on key number 2, **e** is on key number 3, **r** is on key number 4, … , **p** is on key number 10, **a** is on key number 11, … , **l** is on key number 19, **z** is on key number 20, … , **m** is on key number 26.

Keyboard

Input

The first line of the input file `keyboard.in` consists of a single positive integer N. The second line consists of a string with length N, written with lowercase letters of the English alphabet.

Output

On the first line of the output file `keyboard.out` you must write a string, which is a permutation of the letters from a to z. The first letter of the string is written on key number 1, the second – on key number 2 and so on (the original arrangement corresponds to the string qwertyuiopasdfghjklzxcvbnm). On the second line you need to write two integers from 1 to 26, separated by a single space – the numbers of the keys, where Ivancho should initially place his left and right finger respectively. The left finger must be placed on the left side of the right finger.

Scoring

You will receive 0 points if your output is invalid. An invalid output is an output satisfying at least one of the following conditions:

- The string, determining the arrangement of the letters on the keys, is not a permutation of the letters from a to z.
- The starting keys don't satisfy the given condition about their positions.

If your output is valid, you will get $100 * \left(\frac{minScore+1}{yourScore+1}\right)^3$ percent of the points for a given test. We define yourScore as the result received by Ivancho after executing your program and minScore as the smallest result received by Ivancho after executing any program of the contestants.

Constraints

$1 \le N \le 200$

Number of tests	10%	Other 20%	Other 30%	Other 40%
max **N**	50	100	100	200
Max different letters	10	15	20	26

Keyboard

Time limit: 3 sec	Preliminary tests: 20
Memory limit: 256 MB	Final tests: 100

Example test case

Input (keyboard.in)	Output (keyboard.out)
15	iutdjncorepbmyagshkwlxzqvf
iroirutdmyxlubr	1 9

Explanation of the example test case
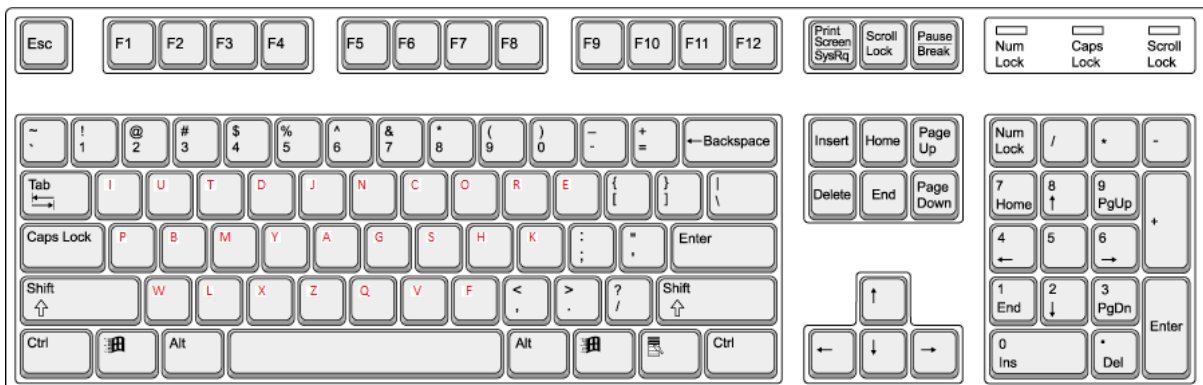
In red we have the new values of the keys.

In the beginning the left finger is on key 1 (I) and the right one is on key 9 (R). We press key (I) with the left finger without moving, press key (R) with the right finger without moving, move the right finger to key 8 (O), press key (I) with the left finger without moving, move the right finger to key 9 (R), move the left finger to key 2 (U), move the left finger to key 3 (T), move the left finger to key 4 (D), move the left finger to key 13 (M), move the left finger to key 14 (Y), move the left finger to key 22 (X), move the left finger to key 21 (L), move the left finger to key 2 (U), move the left finger to key 12 (B), press key (R) with the right finger without moving. Our final result is 482.