Problem 1

Think of a employee record that comprise a name, a employee Id and salary. Name is a string, employee Id and salary is an integer. WAP to define a structure for employee record. Scan number of records in N and list of N employee record into array of structure and output the name and its frequency which is occurring maximum number of times in the list of employee record.

Note: Name does not have any spaces and is in lowercase and Use of structure is mandatory.

Input Format

First line contains integer N which tells the number of employee record then following 3N lines containing list of N employee records in which following each 3 line we have information of employee record i.e. first name then employee id and then salary.

Output Format

Output the name followed by space followed by its frequency (occurring maximum number of times) followed by a newline

Sample Input
4
rahul
1
100
teja
2
120
teja
3
150
abhishek
4
105

Sample Output
teja 2

Problem 2

A magic **square** palindrome is a string whose characters can be divided in a K × K square table with the property that the original sentence can be read from the table in four different ways:

• Start from the (0, 0) cell, move right until the end of the line and then proceed to the next line from (1, 0) cell, move right until the end of line and so on.

• Start from the (0, 0) cell, move down until the end of the column and then proceed to the next column from (0, 1) cell, move down until the end of the column and so on.

• Start from the (K-1, K-1) cell, move left until the beginning of the line and then proceed to the previous line from (K-2, K-1) cell, move left until the beginning of the line and so on.

• Start from the (K-1, K-1) cell, move up until the beginning of the column and then proceed to the previous column from (K-1, K-2) cell, move up until the beginning of the column and so on.

"satorarepotenetoperarotas" is the most famous magic square palindrome. We can arrange it in a K = 5 (5 × 5) table in the following way as shown in figure below (string has 25 characters and the way characters are stored is first 5 character in 0^{th} row and next 5 character in next row and so on…)

s	a	t	o	r
a	r	e	p	o
t	e	n	e	t
o	p	e	r	a
r	o	t	a	s

Notice that the original sentence can be read from the table in the four different ways described above.
Write a program to solve the problem.

Input format: The first line of input gives the number the input strings (without spaces and of maximum 100 length and in lowercase). Then there is following n lines, each containing a string.

Output format: N lines of output, each line has either YES or NO following by newline. (YES if string is a magic square palindrome otherwise NO)

Sample Input:
4
satorarepotenetoperarotas
aba
abba
abcabacba

Sample Output:
YES
NO
YES
NO\n

Note: \n is the newline character.

Explanation:

1st string: You get output as YES since you can exactly divide the strings in characters which can be stored in square of dimension 5 X 5 and also reading from the table in the four different ways gives the same inputted string

2nd string: You get output as NO since you don't have a square table of any dimension which can hold exactly all of its character.

3rd string: Same reason as for 1st string

4th string: You get output as NO since reading from the table (3 X 3) in the four different ways doesn't gives the same inputted string

---

Problem 3:

Encryption is a method to obscure the letters of a sentence to hide its content. It is used in the field of Cryptography to send messages between any two computers securely.

Let there be two persons called Alice and Bob. Alice wants to email a message **M** to Bob. However, she does not want anyone to understand it. So, she uses an encryption algorithm to jumble up the order of letters and convert each letter to a different letter.

A basic version of an encryption algorithm contains **two** steps.

Step A) Rearranging the letter stored in **M** as outlined below. Let the output of this step be message $\mathbf{C_1}$

This is done as follows.

1.) Count the number of letters in the message M. Let this count be N.

2.) Consider the first N Fibonacci numbers (1, 2, 3, 5, 8, 13…). The numbers are in ascending order and denoted hereafter as F. The Fibonacci numbers in F are shuffled in a random order. Let this set of randomly ordered Fibonacci numbers be called R.

3.) Since the number of letters in M and the count of Fibonacci numbers in R is equal, the pair (m_i, r_i) is called a tuple. Here, m_i is the character at index i of the message M and r_i is the number at index i of the set R.

4.) Take a pair (m_i, r_i). Find the position of r_i in the Fibonacci series F. Let this position be *j*. The character m_i will be placed in the j^{th} index of the intermediate output message C_1.

Step B) Each character in $\mathbf{C_1}$ is replaced by a letter some fixed number of positions (*k* position) down the English alphabets. For example, if *k* is *3*, then letter 'a' becomes 'd', 'b' becomes 'e', ..., 'y' becomes 'b', 'z' becomes 'c'. The output of this step is the final encrypted message. **Note:** 'z' becomes 'c' (if *k* = 3). It wraps around in cyclic manner.

Read Sample Input, Sample Output and Explanation to understand step A and B

Your task is to encrypt the message M using the encryption technique described above.

Note:

A) Assume that the first number in Fibonacci series as 1 and the second as 2.

B) Index/position in R and F starts at 0.

Input Format:

The first line is an integer **n** representing the number of letters in the message M (max. of 20 letters in a sentence).

The second line contains **n** Fibonacci numbers separated by space. (Representing R)

The third line is a sentence containing only lower case English alphabets i.e. no spaces and punctuations.

The fourth line is an integer representing the value of k (k varies between 0 and 25).

Output Format:

A single line containing the encrypted message followed by a newline

---

Sample Input:
9
34 13 2 55 3 21 1 5 8
johnisspy
5

Sample Output:
xmnudtxos\n
Note: \n is the newline character.

Explanation: The original message is "john is spy". The message is entered without any spaces as "johnisspy". There are nine letters. The set R [34 13 2 55 3 21 1 5 8] has the first 9 Fibonacci numbers in a random order.
Letter 'j' is mapped to number 34. The position of 34 in the Fibonacci series is 7. Hence, letter 'j' is placed in the 7^{th} index of message C_1. Letter 'o' is mapped to number 13. The position of 13 in the Fibonacci series is 5. Hence letter 'o' is placed in the 5^{th} index of message C_1. Similarly, every other letter is mapped. The intermediate message C_1 is "shipyosjn".
Each letter in C_1 is replaced by a character which is five positions **ahead** in the alphabet (k=5). Letter 's' becomes 'x', 'h' becomes 'm', and so on. The final encrypted message is "xmnudtxos".
