week 5

A. Linear Keyboard

1 second, 256 megabytes

You are given a keyboard that consists of 26 keys. The keys are arranged sequentially in one row in a certain order. Each key corresponds to a unique lowercase Latin letter.

You have to type the word s on this keyboard. It also consists only of lowercase Latin letters.

To type a word, you need to type all its letters consecutively one by one. To type each letter you must position your hand exactly over the corresponding key and press it.

Moving the hand between the keys takes time which is equal to the absolute value of the difference between positions of these keys (the keys are numbered from left to right). No time is spent on pressing the keys and on placing your hand over the first letter of the word.

For example, consider a keyboard where the letters from 'a' to 'z' are arranged in consecutive alphabetical order. The letters 'h', 'e', 'l' and 'o' then are on the positions 8, 5, 12 and 15, respectively. Therefore, it will take $|5-8|+|12-5|+|12-12|+|15-12|=13$ units of time to type the word "hello".

Determine how long it will take to print the word s.

Input

The first line contains an integer t $(1 \le t \le 1000)$ — the number of test cases.

The next $2t$ lines contain descriptions of the test cases.

The first line of a description contains a keyboard — a string of length 26, which consists only of lowercase Latin letters. Each of the letters from 'a' to 'z' appears exactly once on the keyboard.

The second line of the description contains the word s. The word has a length from 1 to 50 letters inclusive and consists of lowercase Latin letters.

Output

Print t lines, each line containing the answer to the corresponding test case. The answer to the test case is the minimal time it takes to type the word s on the given keyboard.

input

```
5
abcdefghijklmnopqrstuvwxyz
hello
abcdefghijklmnopqrstuvwxyz
i
abcdefghijklmnopqrstuvwxyz
codeforces
qwertyuiopasdfghjklzxcvbnm
qqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqqq
qwertyuiopasdfghjklzxcvbnm
abacaba
```

output

```
13
0
68
0
74
```

B. Second Order Statistics

2 seconds, 256 megabytes

Once Bob needed to find the second order statistics of a sequence of integer numbers. Lets choose each number from the sequence exactly once and sort them. The value on the second position is the second order statistics of the given sequence. In other words it is the smallest element strictly greater than the minimum. Help Bob solve this problem.

Input

The first input line contains integer n $(1 \le n \le 100)$ — amount of numbers in the sequence. The second line contains n space-separated integer numbers — elements of the sequence. These numbers don't exceed 100 in absolute value.

Output

If the given sequence has the second order statistics, output this order statistics, otherwise output NO.

input

```
4
1 2 2 -4
```

output

```
1
```

input

```
5
1 2 3 1 1
```

output

```
2
```

C. Boxes Packing

1 second, 256 megabytes

Mishka has got n empty boxes. For every i $(1 \le i \le n)$, i-th box is a cube with side length a_i.

Mishka can put a box i into another box j if the following conditions are met:

- i-th box is not put into another box;
- j-th box doesn't contain any other boxes;
- box i is smaller than box j $(a_i < a_j)$.

Mishka can put boxes into each other an arbitrary number of times. He wants to minimize the number of *visible* boxes. A box is called *visible* iff it is not put into some another box.

Help Mishka to determine the minimum possible number of *visible* boxes!

Input

The first line contains one integer n $(1 \le n \le 5000)$ — the number of boxes Mishka has got.

The second line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$, where a_i is the side length of i-th box.

Output

Print the minimum possible number of *visible* boxes.

input

```
3
1 2 3
```

output

```
1
```

input

```
4
4 2 4 3
```

output

```
2
```

In the first example it is possible to put box 1 into box 2, and 2 into 3.

In the second example Mishka can put box 2 into box 3, and box 4 into box 1.

D. Powers of Two

3 seconds, 256 megabytes

You are given n integers $a_1, a_2, ..., a_n$. Find the number of pairs of indexes i, j $(i < j)$ that $a_i + a_j$ is a power of 2 (i. e. some integer x exists so that $a_i + a_j = 2^x$).

Input

The first line contains the single positive integer n $(1 \le n \le 10^5)$ — the number of integers.

The second line contains n positive integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$.

Output

Print the number of pairs of indexes i, j $(i < j)$ that $a_i + a_j$ is a power of 2.

input

```
4
7 3 2 1
```

output

```
2
```

input

```
3
1 1 1
```

output

```
3
```

In the first example the following pairs of indexes include in answer: $(1, 4)$ and $(2, 4)$.

In the second example all pairs of indexes (i, j) (where $i < j$) include in answer.

E. Mahmoud and Ehab and the message

2 seconds, 256 megabytes

Mahmoud wants to send a message to his friend Ehab. Their language consists of n words numbered from 1 to n. Some words have the same meaning so there are k groups of words such that all the words in some group have the same meaning.

Mahmoud knows that the i-th word can be sent with cost a_i. For each word in his message, Mahmoud can either replace it with another word of the same meaning or leave it as it is. Can you help Mahmoud determine the minimum cost of sending the message?

The cost of sending the message is the sum of the costs of sending every word in it.

Input

The first line of input contains integers n, k and m $(1 \le k \le n \le 10^5, 1 \le m \le 10^5)$ — the number of words in their language, the number of groups of words, and the number of words in Mahmoud's message respectively.

The second line contains n strings consisting of lowercase English letters of length not exceeding 20 which represent the words. It's guaranteed that the words are **distinct**.

The third line contains n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^9)$ where a_i is the cost of sending the i-th word.

The next k lines describe the groups of words of same meaning. The next k lines each start with an integer x $(1 \le x \le n)$ which means that there are x words in this group, followed by x integers which represent the indices of words in this group. It's guaranteed that each word appears in exactly one group.

The next line contains m space-separated words which represent Mahmoud's message. Each of these words appears in the list of language's words.

Output

The only line should contain the minimum cost to send the message after replacing some words (maybe none) with some words of the same meaning.

input

```
5 4 4
i loser am the second
100 1 1 5 10
1 1
1 3
2 2 5
1 4
i am the second
```

output

```
107
```

input

```
5 4 4
i loser am the second
100 20 1 5 10
1 1
1 3
2 2 5
1 4
i am the second
```

output

```
116
```

In the first sample, Mahmoud should replace the word "`second`" with the word "`loser`" because it has less cost so the cost will be 100+1+5+1=107.

In the second sample, Mahmoud shouldn't do any replacement so the cost will be 100+1+5+10=116.

F. Books

2 seconds, 256 megabytes

When Valera has got some free time, he goes to the library to read some books. Today he's got t free minutes to read. That's why Valera took n books in the library and for each book he estimated the time he is going to need to read it. Let's number the books by integers from 1 to n. Valera needs a_i minutes to read the i-th book.

Valera decided to choose an arbitrary book with number i and read the books one by one, starting from this book. In other words, he will first read book number i, then book number $i+1$, then book number $i+2$ and so on. He continues the process until he either runs out of the free time or finishes reading the n-th book. Valera reads each book up to the end, that is, he doesn't start reading the book if he doesn't have enough free time to finish reading it.

Print the maximum number of books Valera can read.

Input

The first line contains two integers n and t $(1 \le n \le 10^5;\ 1 \le t \le 10^9)$ — the number of books and the number of free minutes Valera's got. The second line contains a sequence of n integers $a_1, a_2, ..., a_n$ $(1 \le a_i \le 10^4)$, where number a_i shows the number of minutes that the boy needs to read the i-th book.

Output

Print a single integer — the maximum number of books Valera can read.

input

```
4 5
3 1 2 1
```

output

```
3
```

input

```
3 3
2 2 3
```

output

```
1
```

G. Pangram

2 s., 256 MB

A word or a sentence in some language is called a *pangram* if all the characters of the alphabet of this language appear in it *at least once*. Pangrams are often used to demonstrate fonts in printing or test the output devices.

You are given a string consisting of lowercase and uppercase Latin letters. Check whether this string is a pangram. We say that the string contains a letter of the Latin alphabet if this letter occurs in the string in uppercase or lowercase.

(but you are a mustCPC solver, so solve it using set)

Input

The first line contains a single integer n $(1 \le n \le 100)$ — the number of characters in the string.

The second line contains the string. The string consists only of uppercase and lowercase Latin letters.

Output

Output "`YES`", if the string is a pangram and "`NO`" otherwise.

input

```
12
toosmallword
```

output

```
NO
```

input

```
35
TheQuickBrownFoxJumpsOverTheLazyDog
```

output

```
YES
```