As you all know, Miguel Revilla has been very busy revising the website for the programming contest. But alas, as things go, he lost some files in the upgrade. And even worse: as he backups his files as frequently as most of us, he cannot restore them. He clearly needs some help.

One of the files lost is the file with the test vectors for "Make Palindrome", problem 10453 (http://acm.uva.es/p/v104/10453.html). Miguel needs a program to test the contributions to this problem. In order to do this, he first makes a new file from the file with the test inputs and the file with the outputs of a program: every line consists of the input to the program and its output, separated by a space. Systematic as he is, he makes a list of properties every line in this new file should have:

1. It consists of a first string of lower case characters (length $le1000$), a single space, an integer (≥ 0, ≤ 1000), a single space, and a second string of lower case letters (length ≤ 2000), we will call this property P1.
2. P1 & the second string is a palindrome.
3. P1 & all letters of the first string appear in the second string.
4. P1 & the frequency of every letter in the second string is at least the frequency of this letter in the first string.
5. P1 & the first string can be made out of the second string by removing 0 or more letters (and leaving the order of the letters intact).
6. P1 & the length of the second string is equal to the length of the first string plus the value of the integer.
7. P1 & the value of the integer is smaller than the length of the first string.

Obviously, the list is not complete, but given the time pressure Miguel is under and the short nights he has had during the upgrade, we can easily forgive him.

Input

Every input consists of a concatenation of an input submitted to and an output computed by a program made to the specification of "Make Palindrome", separated by a space. Every line contains less than 5000 ASCII characters. Input is terminated by EOF.

Output

The output consists of the value '`T`' or '`F`' (for True and False; without the quotes) for every property in the list above, and a statement '`The solution is accepted`' if all properties are true, and '`The solution is not accepted`' otherwise (again, both without the quotes). There is a single space between the statement and the '`T`'s and '`F`'s; there is none between the '`T`'s and '`F`'s.

Sample Input

```
abcd 3 abcdcba
aaaa 3 abcdcba
abc 2 abdcba
aab b baab
abababaababababa 0 abababaababababa
pqrsabcdpqrs 9 pqrsabcdpqrqpdcbasrqp
```

Sample Output

```
TTTTTTT The solution is accepted
TTTFFTT The solution is not accepted
TFTTTFT The solution is not accepted
FFFFFFF The solution is not accepted
TTTTTTT The solution is accepted
TTTTTTT The solution is accepted
```