



## 2562 - Dehuff

North America - Greater New York - 2002/2003

A certain data compression technique involves creating a table of variable-length binary codes where one or more binary digits are used to represent a single letter of an alphabet. Usually, letters that occur most frequently in words generated using this alphabet will have shorter binary codes than those used less frequently. For example, in an alphabet consisting of the letters A through Z, in general the letter E appears in more words than the letter Q; therefore it would be expected that E would have a shorter binary code than does Q.

Given a sample string using at least one of each letter in an alphabet, along with the entire binary encoding of that sample string, you should be able to generate at least one binary code table for each letter in the alphabet. For example, consider the sample string: ``CAB" which contains each letter of the alphabet {A,B,C}. If the binary encoding of ``CAB" is ``01011" then the (only) binary code table is:

```
C = 0
A = 10
B = 11
```

The binary codes for each character are *prefix* codes in that no code in the set can be the initial binary string for any other code (so A = 01, B = 011 would not be allowed). For this problem, you will write a program that determines binary code tables for sample strings and their binary encodings. If there is a single binary code table solution, then you will print it out (sorted). If more than one binary code table can be generated from the given set of data, you will print `MULTIPLE TABLES'.

**Note:** For a given alphabet, the entire code space will be used; that is, there will be no unused codes.

### Input

Input consists of a single line consisting of an integer value  $N$  representing the number of datasets that follow. Each dataset consists of two lines. The first line is the sample string that contains at least one of each letter (or space) in the alphabet. The second line is the binary encoding of the sample string.

**Note:** Sample strings may only contain upper case letters and spaces.

### Output

For each dataset, print a line that identifies the dataset in the format: `DATASET # $n$ ' where  $n$  is the dataset number (1 through  $N$ ). If more than one binary code table can be generated to represent the alphabet, print `MULTIPLE TABLES' on a new line and move onto the next dataset. If only one binary code table can be generated, for each character in the alphabet, display the letter, a space, and equal sign `=', a space and the binary code for that letter. Display the table in order sorted by the ASCII value of each character of the alphabet.

## Sample Input

```
3
CAB
01011
HELLO WORLD
1110110111101111011111001111111111110111111011111010
ABCDEFGHI
0101101110111101111101111110001011111111
```

## Sample Output

```
DATASET #1
A = 10
B = 11
C = 0
DATASET #2
= 0
D = 10
E = 110
H = 1110
L = 11110
O = 111110
R = 1111110
W = 1111111
DATASET #3
MULTIPLE TABLES
```

---

Greater New York 2002-2003