Research Project 5:

Huffman Codes (30)

In 1953, David A. Huffman published his paper "*A Method for the Construction of Minimum-Redundancy Codes*", and hence printed his name in the history of computer science. As a professor who gives the final exam problem on Huffman codes, I am encountering a big problem: the Huffman codes are **NOT** unique. For example, given a string "*aaaxuaxz*", we can observe that the frequencies of the characters '*a*', '*x*', '*u*' and 'z' are 4, 2, 1 and 1, respectively. We may either encode the symbols as {'*a*'=0, '*x*'=10, '*u*'=110, '*z*'=111}, or in another way as {'*a*'=1, '*x*'=01, '*u*'=001, 'z'=000}, both compress the string into 14 bits. Another set of code can be given as {'*a*'=0, '*x*'=11, '*u*'=100, 'z'=101}, but {'*a*'=0, '*x*'=01, '*u*'=011, '*z*'=001} is NOT correct since "*aaaxuaxz*" and "*aazuaxax*" can both be decoded from the code 00001011001001. The students are submitting all kinds of codes, and I need a computer program to help me determine which ones are correct and which ones are not.

Input Specification:

Your program must read test cases from standard input.

The input consists of several test cases. For each test case, the first line gives an integer *N* (≤63), then followed by a line that contains all the *N* distinct characters and their frequencies in the following format:

`c[1] f[1] c[2] f[2] ... c[`*N*`] f[`*N*`]`

where c[*i*] is a character chosen from {'0' - '9', 'a' - 'z', 'A' - 'Z', '_'}, and f[*i*] is the frequency of c[*i*] and is an integer no more than 1000. The next line gives an integer *M* (≤1000), then followed by *M* student submissions. Each student submission consists of *N* lines, each in the format:

`c[`*i*`] code[`*i*`]`

where c[*i*] is the *i*-th character and code[*i*] is a string of '0's and '1's.

The input ends with *N* being 0, and that case must NOT be processed.

Output Specification:

For each test case, your program must output to standard output. First print in a line "Case #:" where # is the case number starting from 1. Then in the following *M* lines, either print "Yes" if the student's submission is correct, or "No" if not.

Sample Input:

```
4
a 4 x 2 u 1 z 1
2
a 0
x 10
u 110
z 111
a 0
x 01
u 011
z 001
7
A 1 B 1 C 1 D 3 E 3 F 6 G 6
2
A 00000
B 00001
C 0001
D 001
E 01
F 10
G 11
A 000
B 001
C 010
D 011
E 100
F 101
G 110
0
```

Sample Output:

```
Case 1:
Yes
No
Case 2:
Yes
No
```

Grading Policy:

The report of this assignment is due Monday, May 15th, 2017 at 10:00pm.

- **Programming:** Write the program **(10 pts.) with sufficient comments**.
- **Testing:** Provide a set of test cases to fill in a test report **(3 pts.)**. Write analysis and comments **(3 pts.)**.
- **Documentation:** Chapter 1 **(1 pt.)**, Chapter 2 **(2 pts.)**, and finally a complete report **(1 point for overall style of documentation)**.

The presentation **(10 pts.)** of this assignment is due Tuesday, May 16th, 2017 at 09:50am. All the contributors must be present at the classroom before 09:35am to have the computer ready and the speaker decided.

Peer review of the reports is due Thursday, May 18th, 2017 at 10:00pm.

Final grading sheets will be uploaded after the arbitration.