Senior Division
Duplicates

PROBLEM: A binary search tree becomes quite unbalanced where there are many duplicate keys. Many strategies have been devised to keep binary trees from becoming unbalanced. The strategy that we will try in this program is to insert a single copy of each key into the tree the first time that the key is seen, and maintain with each key the number of times that the key should appear. When trying to add a key that is already in the tree, don't add it! Just increment the counter.

Consider letters in the string ABRACADABRACABOB (duplicate letters are inserted as if they are less than the letter already in the tree). The tree at the left is the standard binary search tree; the one at the right keeps a count of the number of duplicate keys.

INPUT: Five lines of data; each is a string. Ignore all non-letters, and convert all lowercase letters to uppercase. Build the tree shown above, on the right side. That is, add the letters to an initially empty binary search tree, except if the letter is already present in the tree. In that case, increment a counter for that letter.

OUTPUT: Find all nodes with a single child. Print the sum of the counter of all those letters.
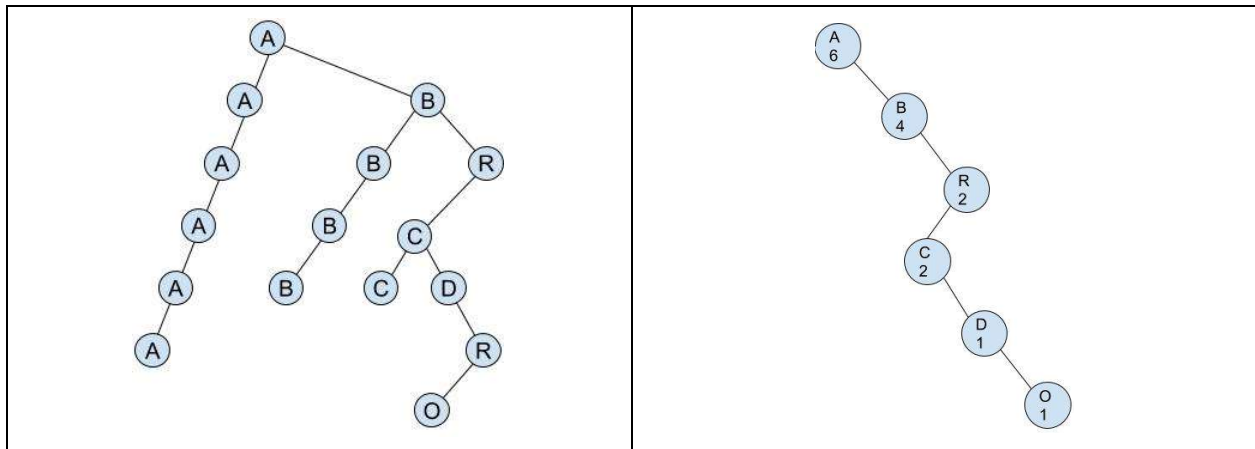
SAMPLE INPUT:
```
abracadabracabob
American Computer Science League
Python and Java are programming languages
Python and Java and java and python
the quick brown fox jumped over the lazy river
```

SAMPLE OUTPUT:
```
1. 15
2. 9
3. 23
4. 18
5. 9
```

TEST DATA

TEST INPUT

simple simon

simple simon simply said something slowly

peter piper picked a peck of pickles

peter piper picked a peck of pickled peppers

how much wood could a woodchuck chuck if a woodchuck could chuck wood

TEST OUTPUT

1. 4

2. 10

3. 7

4. 8

5. 18