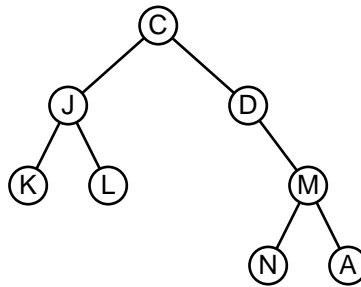


**“BinArrays”**

**Problem:** A *binary tree* is a data structure composed of *nodes*. Each node has three parts: a key (in our case, an uppercase letter), a pointer to a left child, and a pointer to a right child. For example, in the following binary tree



there are 8 nodes: The root of the tree contains the key C. Its left child is the node containing J, and its right child is the node contains D. And so on.

One can use an array to represent a binary tree, by storing the root node at position 1, its left child at 2 ( $2*1$ ), its right child at 3 ( $2*1+1$ ). The children of the node at position  $i$  are stored at  $2i$  and  $2i + 1$ . For example, the tree above is stored in an array as follows:

C	J	D	K	L	-	M	-	-	-	-	N	A	...
---	---	---	---	---	---	---	---	---	---	---	---	---	-----

In this problem, we’ll give you the nodes in a binary tree as a string containing a series of triples. You tell us where in the array key A is stored. For example, the tree above would be represented by the following triples:

MNA CJD JKL D-M

and A is at location 13 in the array representation of the tree.

The triples in a string can appear in any order; you’ll know what the root of the tree is because it’s the only letter that doesn’t appear in the 2nd or 3rd position of a triple.

**Sample Input:**

Line #1: SAL ATR  
 Line #2: HB- BA- ADE  
 Line #3: XW- WJB BCA  
 Line #4: D-G G-H H-C C-A A-X X-Z  
 Line #5: X-Z A-X C-A H-C G-H D-G

**Sample Output:**

Output #1: 2  
 Output #2: 4  
 Output #3: 11  
 Output #4: 31  
 Output #5: 31

**“BinArrays”**

**Test Input:**

Line #1: BCD DAE

Line #2: BCD DEA

Line #3: BCD DEF E-X FGH GA- CJK

Line #4: E-X CJK FGH G-A BCD DEF

Line #5: ABC

**Test Output:**

Output #1: 6

Output #2: 7

Output #3: 28

Output #4: 29

Output #5: 1