Dictionary in C language

To understand how the dictionary works, I make this document, explaining from the structure to all the operations with the binary tree's and list's.

First, each letter has its own node that contains its information, this node is called Letter_T, and its definition is:

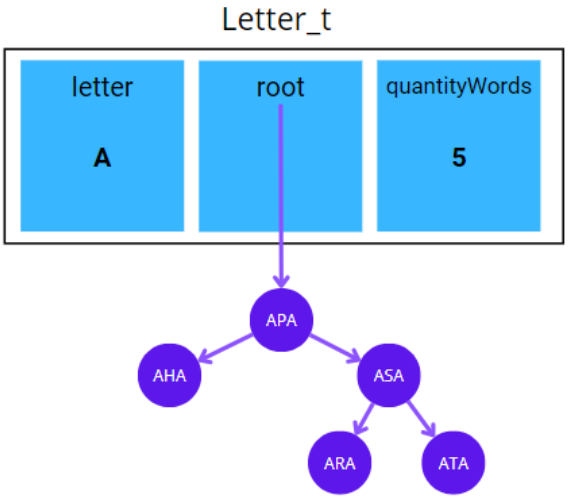
```
typedef struct _node {
   char letter;
   Node_t* root;
   int quantityWords;
} Letter_t;
```

Node_t is another structure, this structure defines a binary tree node, its definition is:

```
typedef struct _nodeT {
   char *word;
   struct _nodeT *right, *left;
} Node_t;
```

As you can see, each letter has its own root node, this node is a binary tree, it contains two children's and a pointer to char to dynamically memory assign.

Graphical reference:



Binary tree operations.

Add nodes:
Max length of a new word is 30 characters, considering it, node's do not always save a 30 characters word, only the length of the word + null character.
First, the position is found recursively and then the node is created.
Words that are the same, are added as left childs.

Delete nodes:
If the node is found, then is deleted recursively, this elimination works like this:
The node is searched again, but with rules, where the children are kept in order not to lose nodes in the process.

Print nodes:
Tree is traversed inorder traversal