Github source code:
https://github.com/nicolagutanu/flcd/blob/main/carina/symbolsTable.py

function insert(self, identifier, pos):
in params: identifier: string -> identifier found in code
Pos: integer -> its index in the code (n th identifier)
out params: none

We traverse the branches of the tree alphabetically (left or right compared to the root) to find where we could fit the new identifier, saving the name of the identifier and it's position in the code. If the identifier already exists in the tree, we return (don't make any changes, the initial position of the identifier in the code is the only one saved).

function search(self, identifier):
in params: identifier: string -> identifier we want to search
out params: pos: string -> position of found identifier or "Not found"

We traverse the tree alphabetically once again while comparing the given identifier to the existing ones in the tree on the left/right branches until we either get to the end of the branch (in which case, the identifier doesn't exist) or until we've found it and returning it's position.

function print(self):
in params: none
out params: none

We traverse the branches in inorder and print the nodes.