

link to GitHub -> https://github.com/915-Pavel-Dragos/FLCD/tree/master/Lab%20_SymbolTable

This Symbol Table is implemented using a hash table with the hash function being the sum of the ASCII values of the characters that form the key modulo the capacity of the symbol table.

The class Node is for encapsulating the functions corresponding to each key - value pair.

The only functions there are in this class are the constructor for creating the named nodes that are going to be added in the hash table

and the `__str__` function for modifying the way nodes look when they are printed.

For the intern representation of the hash table is a list of lists is used the main list being of a fixed capacity, which is initialized

along with the size and capacity field in the constructor of the class.

The insert function uses the hash method to get the index in which list should the new pair be added and then appends it.

This usage of the list of lists also resolves the collisions that may appear as if two pairs have the same hash value they will just be

added to the list that is on the index (in the main list) with the same value as the result of the hash function

The function also increases the size with one.

The search function searched the pair needed by key and value (in case there are multiple pairs with the same value) and then returns

True or False based on whether the pair was found or not.

The remove function searches for a pair, based on its key and value and then it removes it from the list whether the pair was found or not

and it also decreases the size with one.

The `__str__` function modifies the way the SymbolTable looks when it is printed to the console.

The `__len__` function is a function to modify the behaviour of the `SymbolTable` class when it is given as parameter to the `len()` function and

it returns the actual size of the `SymbolTable`.

The `__contains__` function is a function to modify the behaviour of the `SymbolTable` class when it is given as parameter to the `in` function

and by calling the search method it returns `True` or `False`, based on the fact that the pair was found or not.