FLCD Documentation - Acatrinei Raul Cristian 931/1

Github repository:

https://github.com/araulo22/FLCD_datastructure

Symbol table - represented as a hash table, where the symbol table class is just a wrapper for the hash table class. Hash table - python list of dequeues, ensuring that conflicts are solved when hashing two elements to the same position. The hash function computes the sum of all ascii codes of the element, and then returns sum % size, where size is the size of the hashtable.

Hashtable operations:
- add(key) - adds the given key into the table
- remove(key) - removes the given key from the table
- contains(key) - checks if the given key exists in the table, returning a boolean value
- getPosition(key) - returns a pair (a, b) describing the position of the given key
    - a = position in the list
    - b = position in the dequeue

PIF (Program Internal Form):
- a list of pairs (token, position) describing the position of said token in the symbol table
- for operators, separators and reserved words, the position is always (-1, -1) since those are not stored in the symbol table

Tokenizing:
- takes character by character on each line and checks if what we have is:
    * operator
    * separator
    * begins a string
    * constant/identifier
- after that it appends the tokens to a list that is returned

Scanning:
- applies tokenizing on each line
- for each token:
    * constant/identifier -> looks up its position in the symbol table, added to the PIF with "const" or "id"
    * op/sep/rw -> position is (-1, -1)
    * other -> lexical error at that line