

Github link:

<https://github.com/x64alex/Compilers/tree/main/Lab2>

Symbol Table

= a hash table with separate chaining as a resolution method represented as a list of lists.

Fields:

1. size: the number of buckets, private
2. hash_table: the hash table

Methods:

1. init(size: integer) : Initialize an empty hash table with size number of bucket(bucket are the lists which contains lists). Returns nothing.
2. get_size(): Returns the number of bucket (the size that symbol table was initialize with
3. get_hash(key: string): Returns the hashed key. The hash is calculated as the square of the sum of all characters's ascii codes over the length of the key modulo the size of the hash table. The method is private and can be used only inside the symbol table.
4. exists(key: string): Returns true if the key exists, false otherwise.
5. insert(key: string): Inserts the key in the hash_table and returns the bucket number and position where it was inserted. Returns -1 if the value already exists.
6. lookup(key: string): Searches the key in the hash_table and returns the bucket number and position where it is located. Returns -1 if the value does not exists.
7. delete(key: string): If the key exists in the hash table it deletes it and returns the position from where it was deleted. If the key doesn't exists in the hash table it returns -1.