GitHub link: https://github.com/RazvanAndreiLazar/FLCD/tree/main/L2
* works for non-space cases (a=2+1)

HashTable

Representation:
- Array with 509 buckets
- Collision resolution
  closed addressing with dynamic array
- keys are strings

hash function - rolling hash function for strings
p = 51
hash = a[0] p^0 + a[1] p^1 + ... + a[N] p^N % M

search:
compute hash value
search element linearly in the corresponding bucket

insert:
compute hash value
search element linearly in the corresponding bucket
if key exists update the value
otherwise add a new element

SymbolTable → HashTable

index - static field, increasing on each insert, used as a value for insert

Scanner

Fields
file - input file path of the program file
tokens_file - input file path with the programming language tokens
tokens - dictionary token, index with every token from the input file

Methods
set_file - set the file path for the input program

read_tokens - read the tokens from the input file

scan - scan the input program file line by line and parse every one

__parse_line - parse the line word by word and if needed character by character in search for tokens, identifiers and constants

__is_identifier - check if the word is an identifier

__is_constant - check if the word is a constant