Lab 2

Link Github: https://github.com/913-groza-vlad/Formal-Languages-and-Compiler-Design/tree/main/Lab2

My Symbol Table is implemented using a hash table, which can be used as a unique table for both identifiers and constants.

The SymbolTable class is represented as a list (having a specified size) of lists of Strings (these are the symbolic names), the hash table, and it has the following methods:

- the constructor initialize the symbol table with a given size

- private int hash(String key):

This method generates a hash value for a given key/symbolic name by computing the sum of the ASCII codes

of chars of the key in order to place the symbolic name in a specific List in the table and to be easily accessible;

this is achieved applying modulo size to the sum of ASCII codes.

- public boolean insertSymbolicName(String key):

It inserts a symbolic name into the symbol table and returns true if the insertion is successful

and false if the key already exists in the symbol table.

- public boolean containsSymbolicName(String key):

Checks if a given symbolic name exists in the symbol table and returns true if so, false otherwise.

- public Pair<Integer, Integer> searchPosition(String key):

The method 'searchPosition' searches for the position (index) of a symbolic name in the symbol table.

It returns a Pair containing the position of the list in the table and the index in that list at which

the key is found, or (-1, -1) if the name is not found in the table.

- public int getSize():

Getter method for the size field (it returns the size of the symbol table)

- public String toString():

This method provides a string representation of the symbol table for printing

The Pair<K, V> generic class associates two fields of types K and V and

we use it in order to retrieve the position of a symbolic table which is a pair of two integers:

the list position in the table and the index in this list where a symbolic name can be found.

Using the toString() method, a pair can be represented as: (key, value);