https://github.com/alexreitler/FLCD_Lab2

The Symbol table is implemented as a single table for both Identifiers and constants. It uses a single hashtable to store both types of values.

The hashtable implementation uses the method of “buckets”, thus it is made up of a list containing lists. The hash function is the sum of ASCII character codes modulo the [predetermined] size of the table.

addKey(T): Pair<Int, Int> adds an element to the hashtable. After checking if key is not already present in the table, uses the hash function to calculate which list the element will be put in. It is then added to the end of that list. The position of the element is returned as a pair of integers.

addSymbol(String): Pair<Int, Int> – adds the symbol to the table by calling addKey(). The position of the element is returned as a pair of integers.

getPos(T): Pair<Int, Int> returns a pair of integers representing the position of an element in the table.

isSymbol(String) returns true if element already exists in the table, and false otherwise.

getSymbolPosition(String): Pair<Int, Int> returns the position of the symbol in the table as a pair of integers, by calling on the getPos() function.

addKey() and getPos() functions are defined in the HashTable class.

addSymbol(), isSymbol() and getSymbolPosition() functions are defined in the SymbolTable class.