Lab2

https://github.com/915-Nichifor-Dragos/FLCD/tree/master/Lab2/lab-3

Symbol Table:

The Symbol Table is a class composed of three separate hash tables, each designed for specific data types: identifiers, integer constants, and string constants. These hash tables are represented as lists, where each position within the list can store multiple values that hash to the same position. The hash tables have a predefined size. Each element within the Symbol Table is located using a pair of indices, with the first index indicating the list in which the element is stored and the second index indicating the element's position within that list. The hash function used for determining the position of elements in the hash tables differs depending on the data type:

Operations:

addIdentifier(name: string): (int, int) - Adds an identifier to the Symbol Table and returns its position.

addIntConstant(constant: int): (int, int) - Adds an integer constant to the Symbol Table and returns its position.

addStringConstant(constant: string): (int, int) - Adds a string constant to the Symbol Table and returns its position.

hasIdentifier(name: string): boolean - Checks if the given identifier exists in the Symbol Table.

hasIntConstant(constant: int): boolean - Checks if the given integer constant exists in the Symbol Table.

hasStringConstant(constant: string): boolean - Checks if the given string constant exists in the Symbol Table.

getPositionIdentifier(name: string): (int, int) - Retrieves the position of the identifier in the Symbol Table.

getPositionIntConstant(constant: int): (int, int) - Retrieves the position of the integer constant in the Symbol Table.

getPositionStringConstant(constant: string): (int, int) - Retrieves the position of the string constant in the Symbol Table.

toString() (overridden method) - Provides a string representation of the entire Symbol Table.

Hash Table:

The Hash Table is a generic implementation used within the Symbol Table for managing the storage and retrieval of elements. It includes the following operations:

Operations:

hash(key: int): int - Computes the position in the Symbol Table for an integer constant based on the modulo operation with the size of the list.

hash(key: string): int - Computes the position in the Symbol Table for a string constant or identifier based on the sum of the ASCII codes of their characters, followed by a modulo operation with the size of the list.

getSize(): int - Returns the size of the Hash Table.

getHashValue(key: T): int - Returns the corresponding position in the Symbol Table based on the type of the provided 'key.'

add(key: T): (int, int) - Adds the 'key' to the Hash Table and returns its position if the operation is successful; otherwise, throws an exception.

contains(key: T): boolean - Checks if the given 'key' is present in the Hash Table.

getPosition(key: T): (int, int) - Retrieves the position in the Symbol Table of the given 'key' if it exists; otherwise, returns (-1, -1).

toString() (overridden method) - Provides a string representation of the Hash Table.

This structure allows for efficient management and retrieval of identifiers, integer constants, and string constants within the Symbol Table by utilizing separate hash tables for each data type.

Documentation:

https://www.hackerearth.com/practice/data-structures/hash-tables/basics-of-hash-tables/tutorial/

https://research.cs.vt.edu/AVresearch/hashing/strings.php