Git: https://github.com/915-Munteanu-Tudor/flcd/tree/main/lab2

The HashTable class is the implementation of the Hash Table data structure, using the separate chaining method in order to avoid collisions. It uses a list of lists in order to store all the elements with the same hash in the list with its corresponding index equal to their hash. The hash table has the size a prime number because this is how it is recommended in order to improve the hash function. It has implemented more useful methods like:

-contains: returns true or false depending on the existence of the element in the data structure

-str: prints a representation of the Hash Table

-hash: returns the hash of the element consisting of the sum of its ascii characters modulo size of the table

-getPosition: returns None if the element does not exist in the table or a tuple consisting of its position if it exists

-add: returns the position of the element if it already exists, either it adds the element to the Hash Table and returns its position

The SymbolTable class receives a size as argument in order to instantiate a Hash Table as a parameter. It implements the methods: add, contains, str and getPosition, by calling them from the Hash Table class mainly.