The attached Java programs created a customized data structure for storing elements into it in the form of various levels depends on a random value. A summary of the formatting requirements is in the CSCI 3901 course assignment #2 problem #2 information in the course's bright space.

The Java program first reads the method type to choose the random value to be used for further process. Depending on the random value, keys are added to the same level or upper levels.

Program flow:-

- Once the key is read, then the key is searched in the existing levels from top to bottom(if any). If the key is not found then it is added to the lowest level.
- After successful addition, a random value from the Coin class is fetched.
- If the random value is positive then the key is added to the upper level(if any) or creates a new level and gets added to that new level.
- All the records in a level are sorted.
- Sorting order is maintained to have minimum search flows.

Java files:-

- DoublyLinkedList.java (Used for maintaining nodes with upper, below, previous and next references).
- DoublyLinkedListLevels.java (Used for creating and maintaining multiple level nodes with upper level and below level references).
- Node.java (Used for creating data for the given key).
- NodeLevels.java (Used for creating a level with all the keys).
- ListHierarchy.java (Contains multiple functionalities to manipulate data in the levels).
- SkipUI.java (Main class for starting the program).

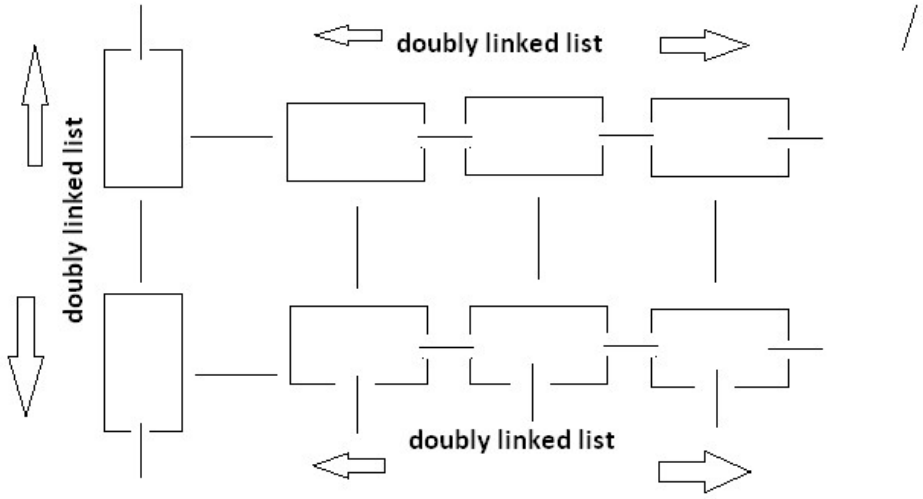
Modules of Program development:-

- Implement a data structure to store keys and various levels of the keys.
- Read key and store the keys in the lower level.
- Check the random value and initiate the process of adding a key to the upper levels.
- Maintain the keys in ascending order to facilitate the search.
- Create a new level if the key is the only element and random value is 1.
- Traverse the levels from the topmost level to the bottom level for finding the element. If not found add at the specific position.
- quit the program.
- Return the boolean values based on successful and unsuccessful operations.

Assumptions:-

- The keys are of String data type.
- Random values are of only 0 or 1.
- Can implement own data structure for maintaining the data.

Data Structure and design:-



Two Doubly linked lists are used for implementing the data structure to maintain and store all the keys and their levels. First, the doubly linked list is used to store keys in the form of nodes. Each node will have its data with an address of next and previous keys and reference to other nodes of the same type in its upper and lower levels.

For the levels, another doubly linked list is used to maintain each level with its keys and reference to the above and below levels.

While adding the keys to a level, it is compared with other keys within the same level and positioned in an ascending approach.

Before adding the key for the first time, it is checked in the level to stop duplicate keys in a level. Used regular expression patterns to check if the key is of only alphabets.

Limitation:-

- Can accept only Strings.
- Keys are not case sensitive.