**IDEA OF IMPLEMENTING SYMBOL TABLE BY 2 METHODS BY TEAM-3**

R. Hasitha sree- AP19110010229

D. Sai Praveen-AP19110010215

K. Vardan-AP19110010211

J. Raghavendra-AP19110010222

M. Devendra sai-AP19110010279

M. George Stephen-AP19110010151

A. Prudhvi raj-AP19110010234

**Implementing through Linear list that is link list:**

We are storing the identifiers in the linked list since we give the input x.text and separating all the lexemes with their respective tokens by using DFA and creating the symbol table in the function called void store\_symb\_tab (char id[]).

Here in this method we are implementing same technique for separating the lexemes and for symbol tables we use node for interchanging the address and data for storing identifier.

In link list a pointer “HEAD” is maintained to point to first record of the symbol table.

If head=NULL the data is assigned.

If head is not equal to NULL then it checks data is recurred or not.

If the identifiers are present, then it returns the new node or it returns to its original state.

**Implementing through Linear list that is Hashing list:**

In hashing 2 tables are maintained a hash table and symbol table and is the most commonly used method to implement these symbol tables.

A hash table is an array with index range of 0 to table size -1, the entries are pointers pointing to names of symbol table.

For searching a name, we use hash function that will result in any integer between 0 to table size-1

Advantage of quick search is possible and disadvantage is that hashing is complicated to implement.

Implementation is similar to link list and while implementing through arrays the key collision may occur so we are implementing using link list.