Name: Saurabh Mukherjee Roll no: 001910501006 Class: BCSE II Sem: First Session 2020-21 Assignment Set :2

Problem No. 3

Problem Statement :

Define an ADT for List. Write C data structure representation and functions for the operations on the List in a Header file with array as the base data structure. Write a menu-driven main program in a separate file for testing the different operations and include the above header file. Two data structures with and without using sentinels in arrays are to be implemented.

Solution Approach:

An array is to be created where we are going to store the values of the list .We keep a sentinel value of -1 to denote the end of the array. Whenever we add values to the list we increment append the value to the current sentinel position and shift the sentinel .The array size is taken to be large to avoid the functionality of adding as many nodes as the user wants as in case of the linked list .While deleting a particular value we shift the complete list after that particular node to a position of left .While printing we print the list until we have the sentinel reached .

In case of not using a sentinel we can keep a length variable at any point of time to get the length of the list created using the array and do the further operations required to be performed.

Structured Pseudocode :

1.initialise a list structure which contains an array say array[MAX]

2.//To add an element

3. for i from 0 till l.array[i]!=-1

4. l.array[i]=element,l.array[i+1]=-1;

5.//search a node for a given value say val

6.traverse the array till l.array[i]!=-1

7. if val==l.array[i]

8. return found

9. return not found

10.//delete a node with the value say val

11for i from 0 to l.array[i]!=-1

12. if (l.array[i]==val)

13. for j from i to l.array[j]!=-1

14. l.array[j]=l.array[j+1]

15.//search a particular node for a value say val

16.for i from 0 to l.array[i]!=-1

17. if (l.array[i]==val),return found

18.return not found

19.//delete the front node

20.call del\_node() function for the value of first element

Results:

We are declaring array as the base date type in this case to implement the list date structure .We need to store the list elements index wise in the array and whenever any node is inserted or deleted the corresponding function is executed .

The traversal of the list ,insertion deletion ,search ,deletefront all take linear algorithmic time to perform the operations.

Discussion:

We need to ensure that the size of the array is sufficient to hold a large number of values because in the list date structure the user can add as many nodes as he wants , so the array data structure used to implement the list too, should ensure the same thing .In case where we have a sentinel we get the end of the list easily but in case where we don’t need to use the sentinel we must be taking length to determine the length of the list at any point of time .

Separate files containing commented source code

The file has been attached.