**UCS 2312 Data Structures Lab**

**Exercise 3: Doubly Linked List and its applications**

Create an ADT for the doubly linked list data structure with the following functions. Each node which consists of integer data, address of left and right nodes [CO1, K3]

Create a ListADT which has implementations for the following operations

1. Insert an item in the front of the list void insertFront(listADT L, int c)
2. Display the items from the list void displayItems(listADT L)
3. Delete the item present in the list void deleteItem(listADT L, int c)
4. Search an element in the list and return the number of occurrences int searchItem(listADT L, int c)

Write a program in C to test the ListADT for its operations with the following test cases.

Testcase:

Initially L is Empty insertFront(L,6) → header→6

insertEnd(L,2) → header→2→6 insertMiddle(L,2,1) → header→2→1→6

insertMiddle(L,2,1) → header→2→1→1→6 search(L,1) → 2

In addition, do the following operations:

1. Delete the nodes with odd numbers in the list.
2. Replace the nodes with odd sum digit values with random odd numbers
3. Add a single digit to a DLL.

Example: Input:

header→1→9→9→9

digit =6

Output: header→2→0→0→5

Input:

header→1→9→9→3

digit =6

Output: header→1→9→9→9

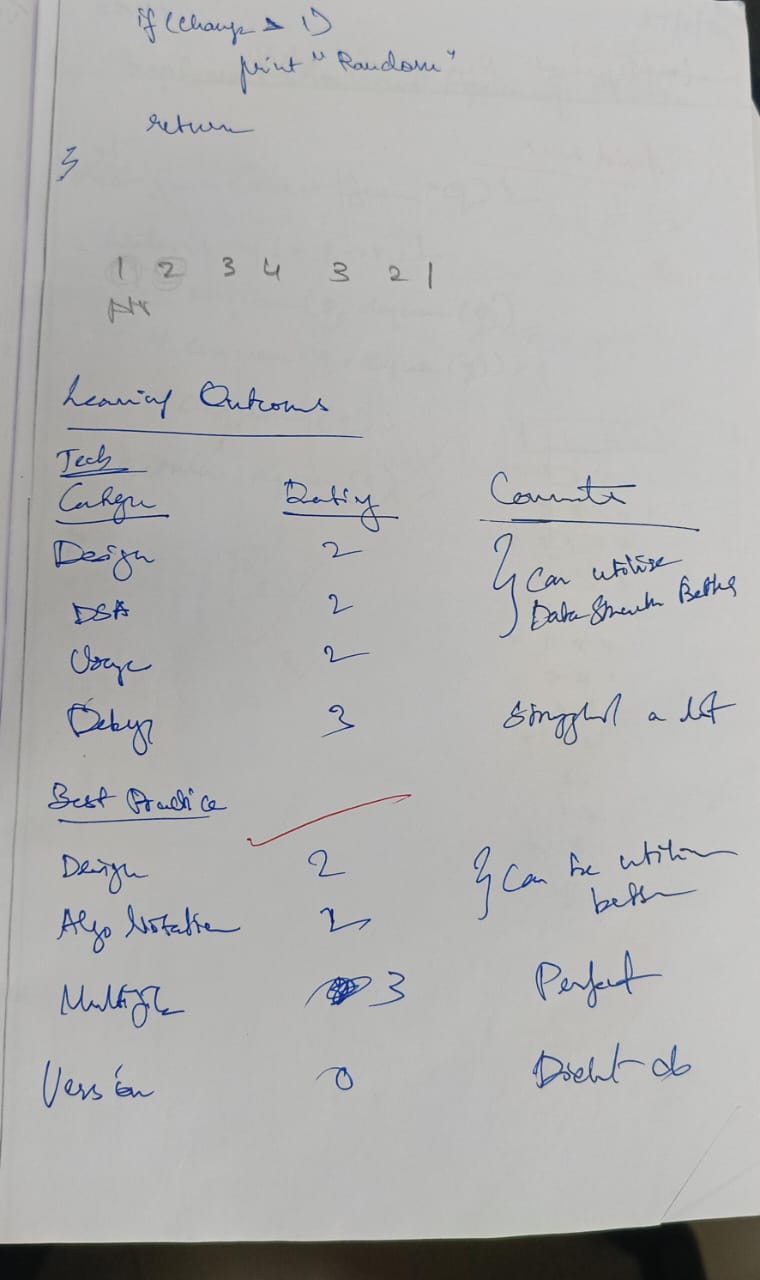
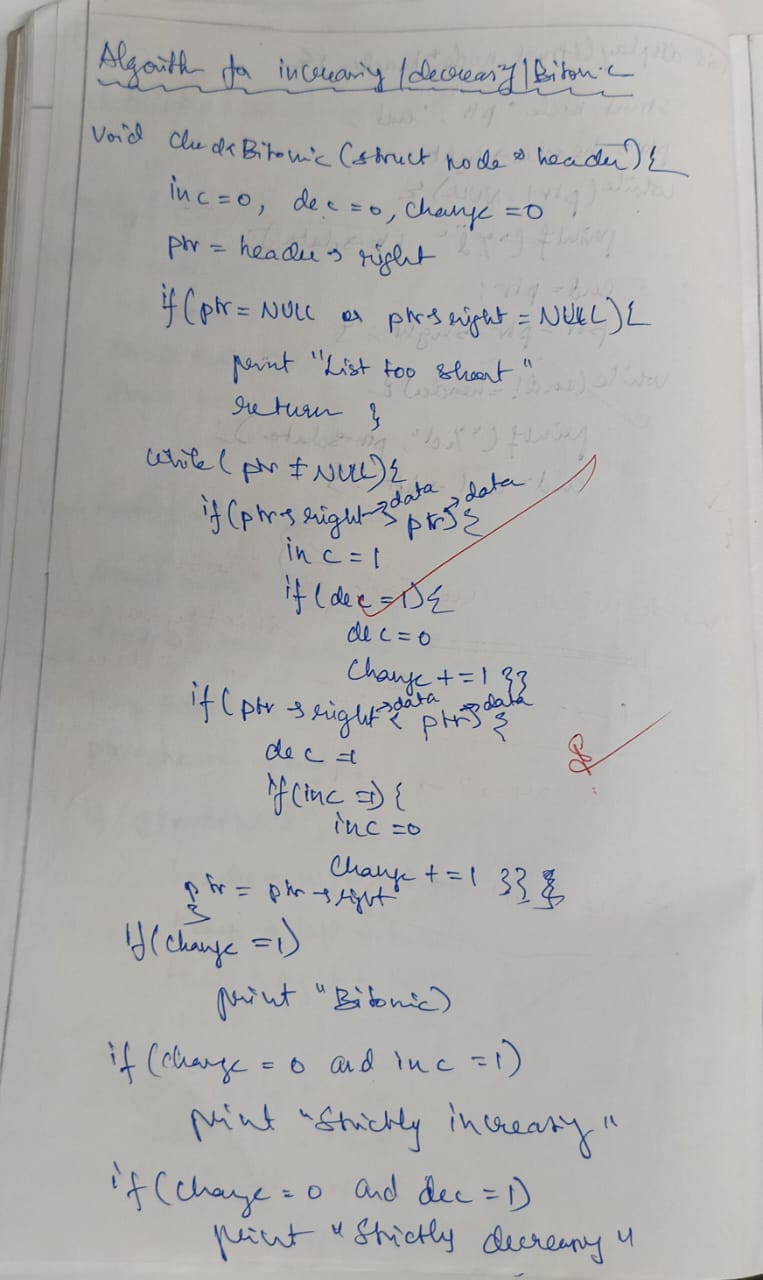
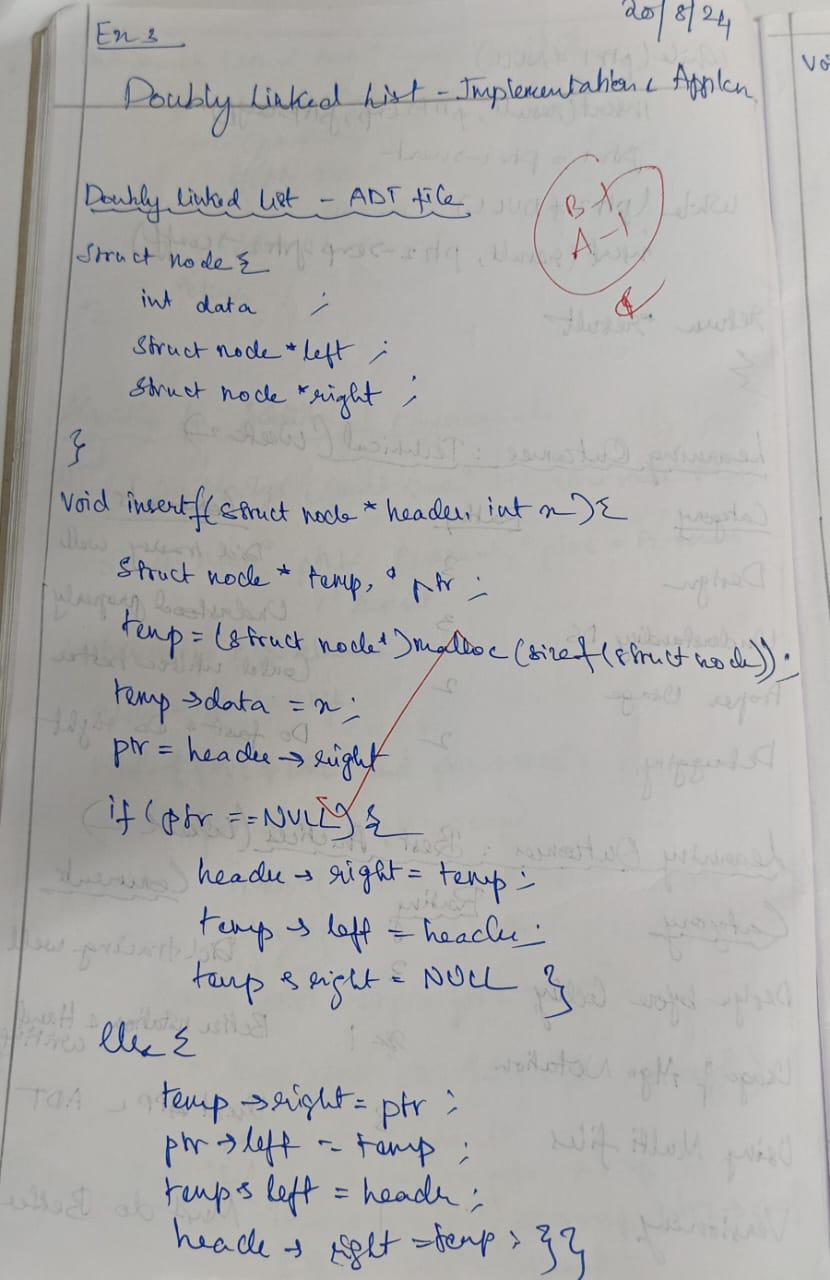
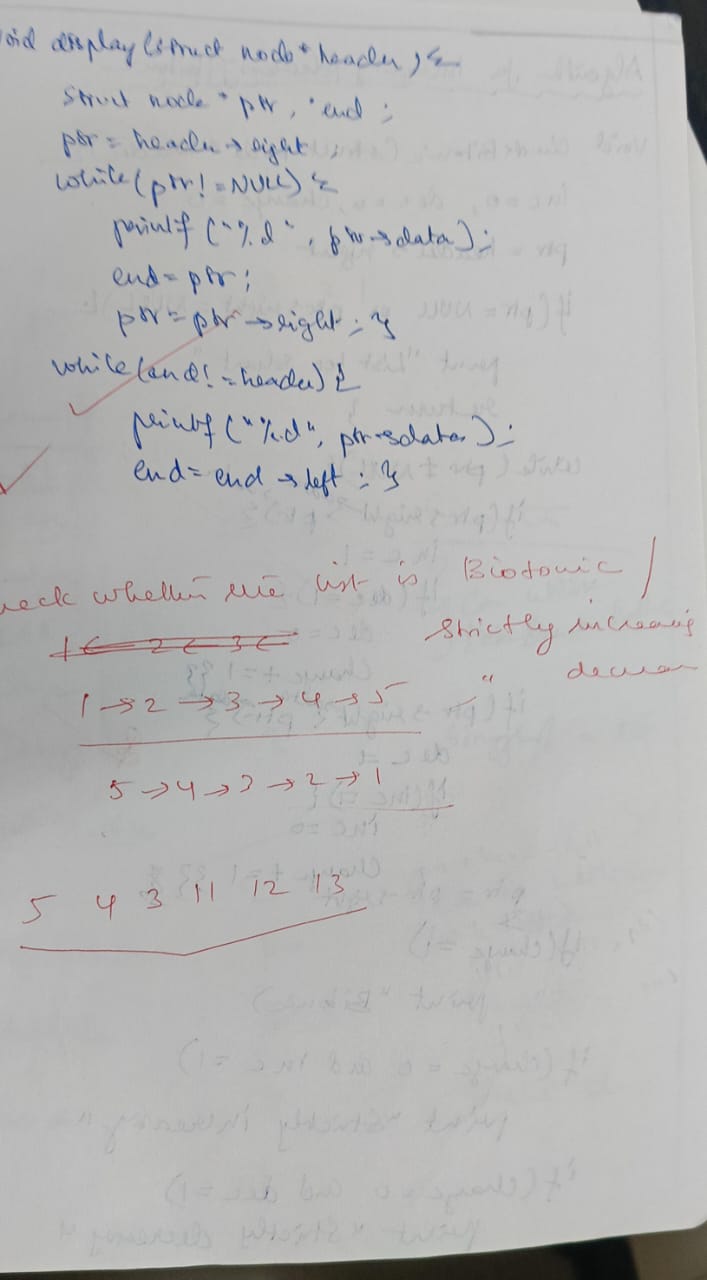
Input:

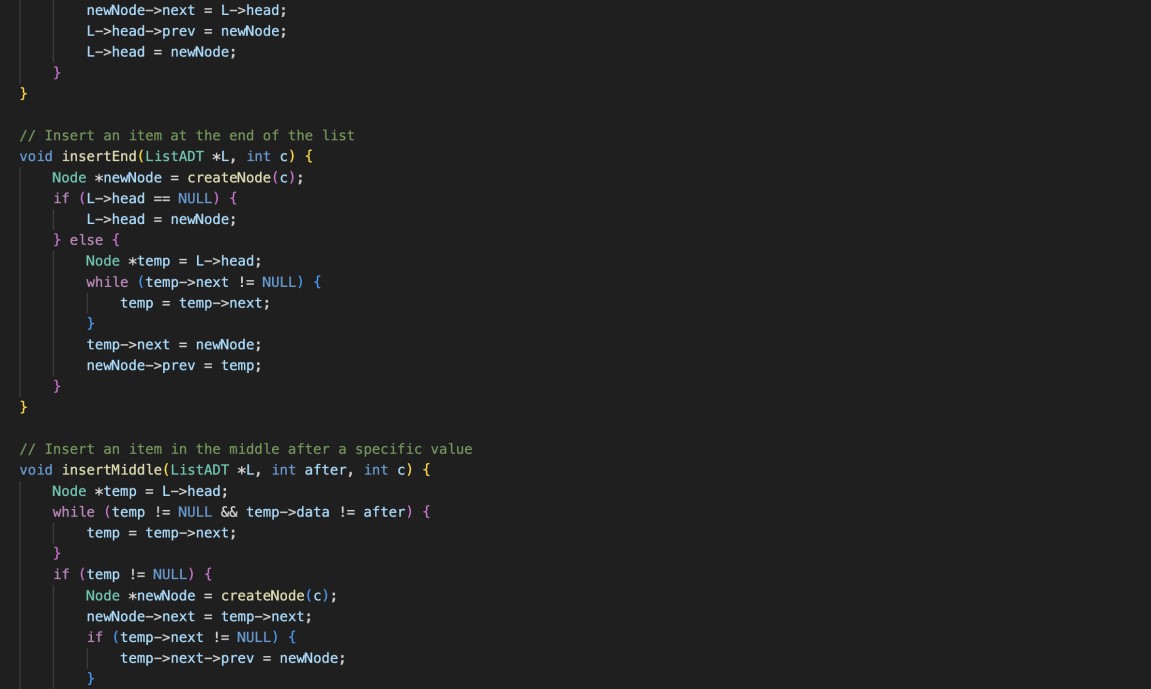
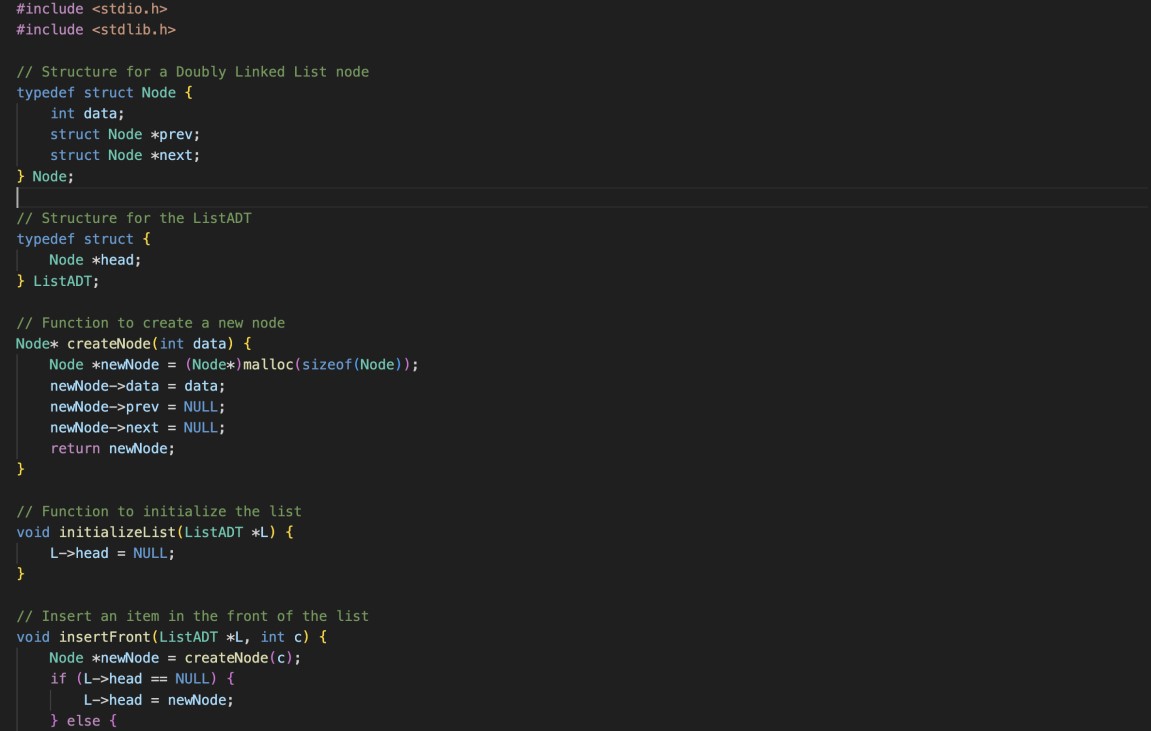
header→9→9→9→9

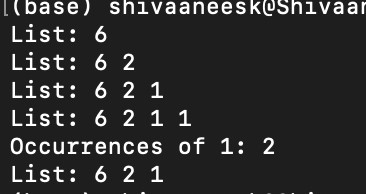
digit =6

Output:

header→1→0→0→0→5







Technical Outcomes

|  |  |  |
| --- | --- | --- |
| Design | 2 | Needs improvement |
| Understanding of DS | 2 | Needs improvement |
| Use of DS | 2 |  |
| Debugging | 3 |  |

Best Practices

|  |  |  |
| --- | --- | --- |
| Design before coding | 2 | Needs improvement |
| Usage of Algo | 2 |  |
| Multifile | 3 |  |
| Versioning | 0 |  |