

**Ex : 11**

**Title:** Design and develop a Program to maintain a dynamic list of recently returned books in a library.

**Problem Description:** To simplify management, the librarian uses a header linked list, where the header node stores metadata about the list.

**Method:** The program must support the following tasks:

1. When a book is returned, its ID should be added at the beginning of the list.
2. The librarian frequently wants to know how many books are currently in the returned-books list, so your program must count and display the total number of nodes using the header node.
3. Sometimes a book at the front of the list is immediately taken for shelving. In such cases, your program must delete the book at the beginning of the list.
4. The librarian should be able to search for a specific book ID to check whether it is present in the returned-books list.

**Theory Reference:** Module 3

***Algorithm:*****Insert at Beginning (book\_insert)**

Input: book ID to be inserted

Output: Node inserted at the beginning

1. Create a new node ptr.
2. If memory allocation fails, print "Overflow" and stop.
3. Read the book ID into item.
4. Set ptr->data = item.
5. Set ptr->next = head->next.
6. Set head->next = ptr.
7. Increment head->data (node count).
8. Print "Node inserted".

**Algorithm: Delete from Beginning (book\_delete)**

Input: —

Output: First node deleted, so that it can be put in shelf

1. If head->next == NULL, print "List is empty" and stop.
2. Set temp = head->next.
3. Set head->next = temp->next.
4. Free the node temp.
5. Decrement head->data (node count).
6. Print "Node deleted".

**Algorithm: Search (search)**

Input: book ID(key) to search

Output: Location of key or "Not found"

---

1. If head->next == NULL, print "Empty List" and stop.
2. Read the search key into key.
3. Set temp = head->next and pos = 1.
4. Repeat while temp != NULL:
  - o If temp->data == key, print "Found at position = pos" and stop.
  - o Else move temp = temp->next and increment pos.
5. If loop ends, print "Book ID not found".

**Algorithm: Display (display)**

Input: —

Output: All returned book IDS printed

1. If head->next == NULL, print "List is empty" and stop.
2. Set temp = head->next.
3. While temp != NULL:
  - o Print temp->data.
  - o Move temp = temp->next.
4. Stop.

**Output:**

1. Insert book in beginning
2. Delete book from Beginning
3. Search for a book ID
4. Show all returned book iDS
5. Total returned books
6. Exit

Enter your choice?

1

Enter book ID101

*Node inserted*

1. Insert book in beginning
2. Delete book from Beginning
3. Search for a book ID
4. Show all returned book iDS
5. Total returned books
6. Exit

Enter your choice?

1

Enter book ID102

*Node inserted*

1.Insert book in begining  
2.Delete book from Beginning  
3Search for a book ID  
4.Show all returned book iDS  
5.Total returned books  
6.Exit  
Enter your choice?  
1

Enter book ID103

Node inserted  
1.Insert book in begining  
2.Delete book from Beginning  
3Search for a book ID  
4.Show all returned book iDS  
5.Total returned books  
6.Exit  
Enter your choice?  
2

Node deleted from the begining ...

1.Insert book in begining  
2.Delete book from Beginning  
3Search for a book ID  
4.Show all returned book iDS  
5.Total returned books  
6.Exit  
Enter your choice?  
4

printing IDS of books . . . .

102  
101  
1.Insert book in begining  
2.Delete book from Beginning  
3Search for a book ID  
4.Show all returned book iDS  
5.Total returned books

---

6.Exit

Enter your choice?

5

Total number of nodes 2

1.Insert book in begining

2.Delete book from Beginning

3Search for a book ID

4.Show all returned book iDS

5.Total returned books

6.Exit

Enter your choice?

3

Enter book ID which you want to search?

101

Book ID found at location 2

1.Insert book in begining

2.Delete book from Beginning

3Search for a book ID

4.Show all returned book iDS

5.Total returned books

6.Exit

Enter your choice?

6

---

-----  
Process exited after 41.63 seconds with return value 0

Press any key to continue . . .