 **Bharatiya Vidya Bhavan’s**

**SARDAR PATEL INSTITUTE OF TECHNOLOGY**

(Autonomous Institute Affiliated to University of Mumbai)

Munshi Nagar, Andheri (W), Mumbai – 400 058.

COMPS Department

|  |  |
| --- | --- |
| **Experiment** | 1 |
| **Aim** | **perform following operations on doubly linked lists 1. Create 2. find the length of list using recursion 3. delete a node 4. display** |
| **Name** | David Daniels |
| **UID** | 202330038 |
| **Class** | Div -A |
| **Batch** | C |
| **Date of Submission** | 10-9-24 |

|  |  |
| --- | --- |
| **Theory** |  |
| **Algorithm** |  |
| **Problem Solving** |  |
| **Program(Code)** | #include <stdio.h>  #include <stdlib.h>  #include <stdbool.h>  typedef struct node  {      int data;      struct node \*next;      struct node \* prev;  }node;  void display (node \* head)  {      node \* temp = head;      printf("\n");      while (temp->next!=NULL)      {          temp=temp->next;          printf("%d ->", temp->data);      }      printf(" NULL \n");      printf("\n\n");  }  int length (node \* head)  {      if (head==NULL)      {          return 0;      }      return (1+length(head->next));  }  node \* get\_node(int data)  {      node \* new = (node \* ) malloc(sizeof(node));      new->data=data;      new->next=NULL;      new->prev=NULL;      return new;  }  node \* insert\_at\_start (node \* head , int data)  {      node \* new = get\_node(data);      head->next->prev=new;      new->next=head->next;      head->next=new;      display(head);      return head;  }  node \* insert\_at\_end (node \* head , int data)  {      node \* new = get\_node(data);      node \* temp=head;      while (temp->next!=NULL)      {          temp=temp->next;      }      temp->next=new;      new->prev=temp;      display(head);      return head;  }  // delete at end , left specific  node \* delete\_at\_end (node \* head )  {      node \* temp=head;      while (temp->next!=NULL)      {          temp=temp->next;      }      //temp is now last node;      temp->prev->next=NULL;      free(temp);      display(head);      return head;  }  node \* delete\_at\_left\_of\_data(node \* head , int value)  {      node \* temp=head;      while (temp->next!=NULL && temp->data!=value)      {          temp=temp->next;      }      node \* to\_del = temp->prev;      //temp is now the node      // delete the node to the left of it      temp->prev->prev->next=temp;      temp->prev = temp->prev->prev;      free(to\_del);      display(head);      return head;  }  void menu (node \* head)  {      int choice=0;      printf("Welcome to choice \n Enter 1 to print \n Enter 2 to insert at start \n Enter 3 to insert at end \n Enter 4 to delete at end \n Enter 5 to delete at left of data \n Enter 6 to print length of Doubly LL \n Enter 0 to end\n");      int num=0,len=0;      while (true)      {          printf("Enter choice \n");          scanf("%d",&choice);          if (choice==0)          {              printf("Goodbye");              break;          }          switch (choice)          {          case 1:              display(head);              break;          case 2:              printf("Enter number to insert at start ");              scanf("%d",&num);              insert\_at\_start(head,num);              break;          case 3:              printf("Enter number to insert at end ");              scanf("%d",&num);              insert\_at\_end(head,num);              break;          case 4:              delete\_at\_end(head);              break;          case 5:              printf("Enter number to delete at left of ");              scanf("%d",&num);              delete\_at\_left\_of\_data(head,num);              break;          case 6:              len = length(head)-1;              printf("Length is %d \n\n", len);              break;          default:              printf("Error: ");              break;          }      }  }  int main(int argc, char const \*argv[])  {      node \* head = (node \* ) malloc(sizeof(node));      int num=0;      printf("Enter 1 manditory node: ");      scanf("%d", &num);      node \* n1 = get\_node(num);      head->next=n1;      menu(head);      return 0;  } |
| **Output** |  |
| **Conclusion** | Thus I have learned how to implement doubly linked lists in C and how to perform various insertions and deletions in the same.  I have also learned how to use recursion in a linked list. |