Date 16-12-24
Page

7 WAP to implement doubly link list wi
-th primitive operations.
a) create a doubly linked list
b) insert a new node to the Left of t
node
c) Delete the node based on a specific
value
d) Display the contents of the List

```
#include<stdio.h>
# include <stdlib.h>

 struct node{
  int data;
  struct Node*Prev;
  struct Node*Next;
  };

 struct Node* createNode (int value){
  struct Node* nNode = (struct Node*) malloc (size
-of (struct Node));
  newnode->data = value;
   newNode->Prev = NULL;
   newNode -> next = NULL;
    return newNode;
 }
 void insertLeftof ( struct Node* head, int targ
-et, int value) {
  struct Node* current =head;
  while (current != NULL && current -> data!= target)
   {
    current = current->next;
   }
```

```
if (current == NULL)
{
 print ("Node with value %d not foun
 target);
 return;
}
  struct Node* newNode = CreateNode (valu
  newNode -> next = current
  newNode -> prev = current -> prev;

  if (current -> prev != NULL) {
     current -> prev -> next = newNode;
  }
  else {
       head = newNode;
     }
  current -> prev = newNode;
  print ("Inserted %d to the left o
  %d \n", value, target);
 }

 void deleteNode (struct Node* head, int
  struct Node* current = head;
  while (current != NULL && current -> data !=
       current = current -> next;
    }
  if (current == NULL)
  {
   print ("Node with value %d not fo
    found. \n", value);
    return;
  }
```

Date 16-12-24
Page

```
if(current ->prev != NULL)
 {
 current ->prev->next = current->next;
 }
 else{
  head = current->next
  }
 if(current->next != NULL)
  { current ->next ->prev = current -> prev;
  }
  free(current);
  printf("Deleted node with value %d\n",val
   ue);
 }
 void display (struct node* head) {
     if(head==NULL)
       { printf("List is empty.\n");
          return;
        }
     struct Node* current = head;
     Printf("Doubly linked list:");
     While(current!=NULL){
       printf("%d ", current->data);
       current = current-> next;
       }
      Printf("\n");
  }


  int main()
  {
   Struct Node*head = NULL;
    int choice, value, target;
```

```
while(choice) {
  case 1:
        printf("Enter the target value to
insert left of:");
        scanf("%d",&target);
        printf("Enter the value to insert
        scanf("%d", & value);
        insertLeftof (head, target, value);
        break;
  case 2:
        printf("Enter the value of the
        -de to delete:");
        scanf("%d", &value);
        deleteNode(head, value);
        break;
  case 3:
        display(head);
        break;
  case 4:
        printf("Exiting program...\n");
        exit(0);
  default:
        printf("Invalid choice. Try again
      }
    }
   return 0;
}
```

Date 16-12-24
Page

O/P

```
Doubly Linked list operations
1. Insert Left of node
2. Delete Node by value
3. Display List
4. Exit

Enter your choice:1
 Enter the target value to insert Left o
 10
 Node with value 10 not found.

Enter your choice:1
 Enter the value of the node to delete:
 Node with value 5 not found

Enter your choice:3
 List is empty

Enter your choice:4
 Exit.
```