05 02 2024

1. WAP to implement doubly link list with primitive operations.

a) Create a doubly linked list.
b) Insert a new node to the left of the node.
c) Delete the node based on a specific value.

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

struct node
{
 int data;
 struct node *next;
 struct node *prev;
};

struct node *head;

void create_dll()
{
 struct node *new_node, *ptr;
 int num;
 printf(" Enter -1 to exit . \n");
```

```
while (num != -1)
{
    printf ("Enter the num: ");
    scanf ("%d", &num);
    new_node = (struct node *) malloc (sizeof(struct node));
    new_node -> data = num;
    if (head == NULL)
    {
     head = new_node;
     new_node -> next = NULL;
     new_node -> prev = NULL;
    }
    else
    {
     ptr = head;
     while (ptr -> next != NULL)
     {
        ptr = ptr -> next;
     }
     ptr -> next = new_node;
     new_node -> prev = ptr;
     new_node -> next = NULL;
    }
}
}
```

```
void ingeard - defd ()
{
  sdaved node *newnode, *ptar;
  ind val, num;
  newnode = (sdaved node*) malloc (sizeof (sdaved node));
  paindf(" Enter a value to ingeard at defd: \n");
  seanf(" %d ", &val);
  paindf(" Ender the val of node: ");
  seanf("%d", &num);

  newnode -> data = val;
  ptar = head;
  while (ptar -> data != num)
  {
    ptar = ptar -> next;
  }.
  ptar -> pre // -> next = newnode;
  newnode -> prev = ptar -> prev;
  newnode -> next = ptar;
  ptar -> prev = newnode;
}.
```

```
void display()
{
  struct node *ptr;
  if (head == NULL)
  {
   printf ("LL is empty");
  }
  else
  {
    ptr = head;
    while (ptr -> next != NULL)
    {
        printf ("%d", ptr -> data);
        ptr = ptr -> next;
        }
  }
}

void del()
{
   struct node *ptr;
   int value;
   printf ("Enter the value to be deleted: ");
   scanf ("%d", &val);
   ptr = head;
```

```
if (head -> data == val)
{
    ptor = ptor -> next;
    head = ptor;
}
else
{
  while (ptor -> data != val)
  {
   ptor = ptor -> next;
  }
  ptor -> prev -> next = ptor -> next;
  ptor -> next -> prev = ptor -> prev;
}
 free(ptor);
}
```

Output:

---- Menu ----

1. Concate LL
2. Ingeart - left
3. delete
4. diggplay
5. exit

1

Enter -1 to exit

Enter the num: 1

Enter the num: 2

Enter the num: 3

Enter the num: 4

Enter the num: -1

4

1 -> 2 -> 3 -> 4

Sun
5/2/24

2

enter the val to ingeart d left: 99

enter the val of node: 46

4

1 -> 99 -> 2 -> 3 - 4

3

Enter the value to be deleted: 99 4 -1 -> 2 -> 3 -> 4