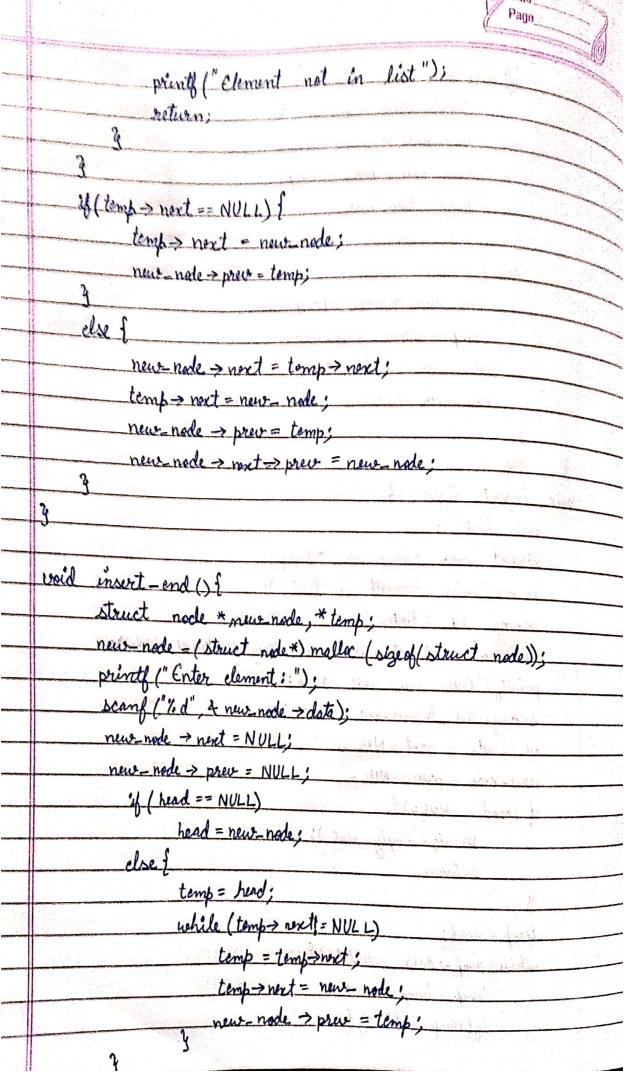
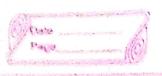
	(D) bate
	LAB 9
	# include < stdie. h)
	# include < stdlib.h)
And the second second	
or state of the st	struct node {
	int data;
	Struct node *next;
	struct node * prev;
Armed American Company of the Compan	3;
	struct node * head = NULL;
	void insert-left () of
	int listele;
	struct node * new_node, * temp;
	printf ("Enter element in list: ");
	1 "" 4 1" 2 1: tolo
٠٠٠-	new node = ( struct node *) mallac ( size of ( struct + male);
	printf ("Enter new noode data: ");
-4-	sconf ("%d", & new-node > data);
	new-node -> next = NULL;
<u> </u>	A. > Dease- NULLS
	if (head == NULL) { hasaid
	prints ("Empty list");
	return;
	newin,
	temp = head; while (temp -> data! = listele) {
<u></u>	while (temp > aara ; - mart!
···	temp = temp → next;
سون	if (temp == NULL) {  printf("Element not in list");
<b>,,,</b>	prints ( Element not but ties )
	return;
A CONTRACTOR OF THE PARTY OF TH	



```
if temp = prece = NULL) f
       timp - prev - new-nucle;
         now note > pres = NULL:
          new-nated next e temp;
          head - new node:
     the f
        new note > prev = temp > prev:
        temb > prece = new node;
        new-note > next = temp;
       new mode > prev > rock = new mode;
void insert-right () {
   struct node * new-node, * temp;
   printf ("Enter element in list");
   scanf (" " d' A listele);
  new-node = (struct node *) malloc (size of struct node)):
   printf ("Enter new node data;");
   scanf(""/d", + new node >data);
  new-node > next = NULL;
  new node > prev = NULL;
  if (head == NULL) {
    printl ("Empty list");
       return:
  temp = head;
  while (temp > data |= listile) {
      temp = temp > next;
     4 (temp = NULL) 1
```





```
wid Alos
       street not thesi
       int de
       4/ And = = NOLD ]
            printf ('Empty List'):
       printf ("Enter elevent to delete: "):
       scanff" "Ad", tele):
       temp=head:
       while (temp -> date != ele){
            temp= temp> next;
           if (temp== NULL) {

printf ("Clement not in list"):
                socielistica return;
       if (temp== head)
              head = bead > met:
       elx if (temp > next == NVLL) }
              temp = temp = prevs
             temp > wet = NULL;
           temp → pseu → next = temp → next;
           temp > rext > prev = temp > prev;
       3
void display () {
     struct node * temp;
     temp = head;
```

