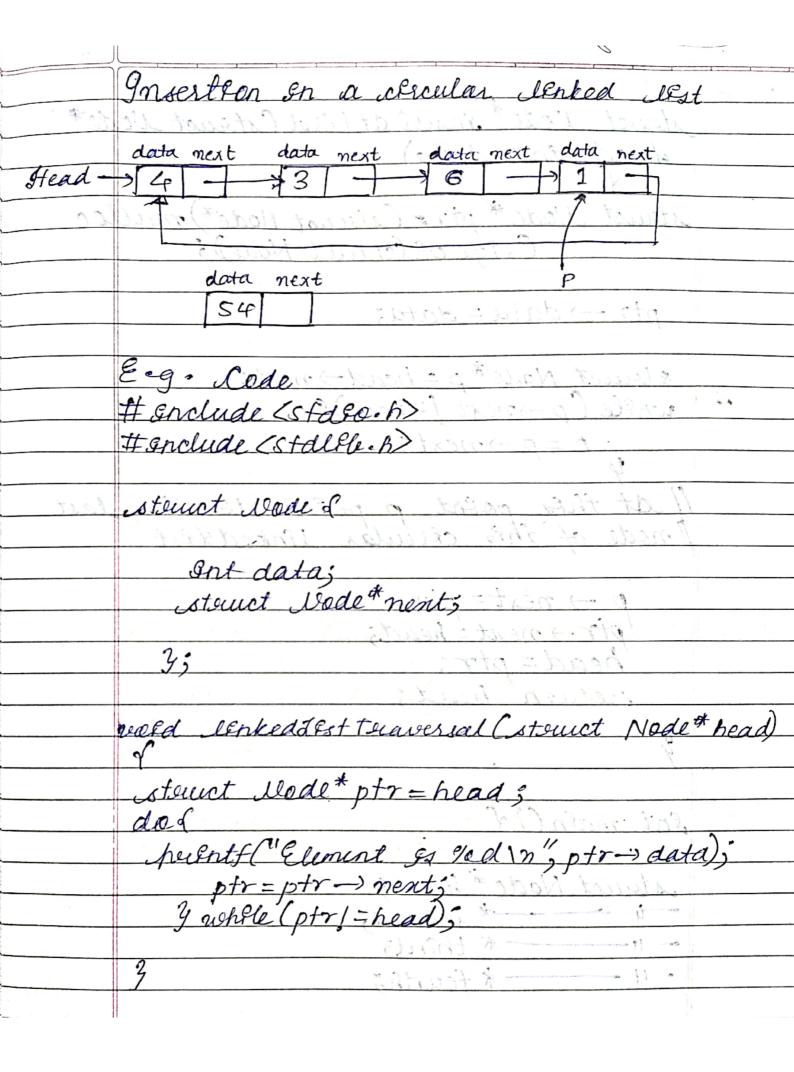
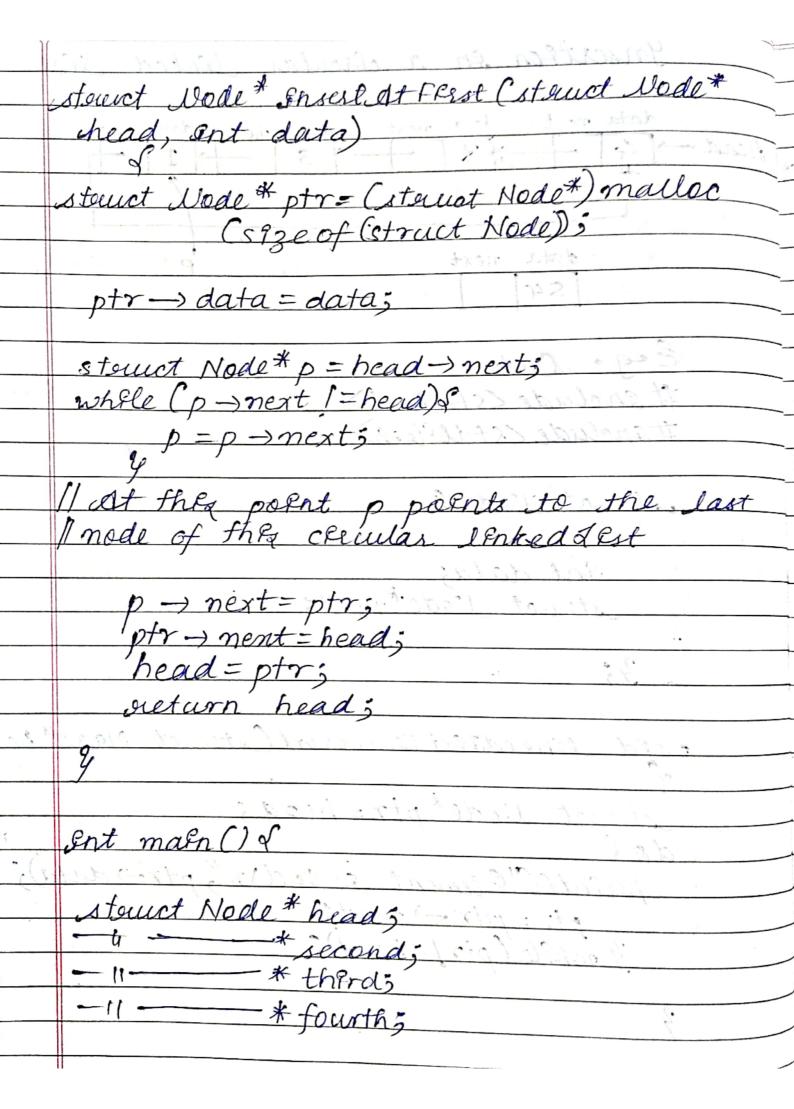
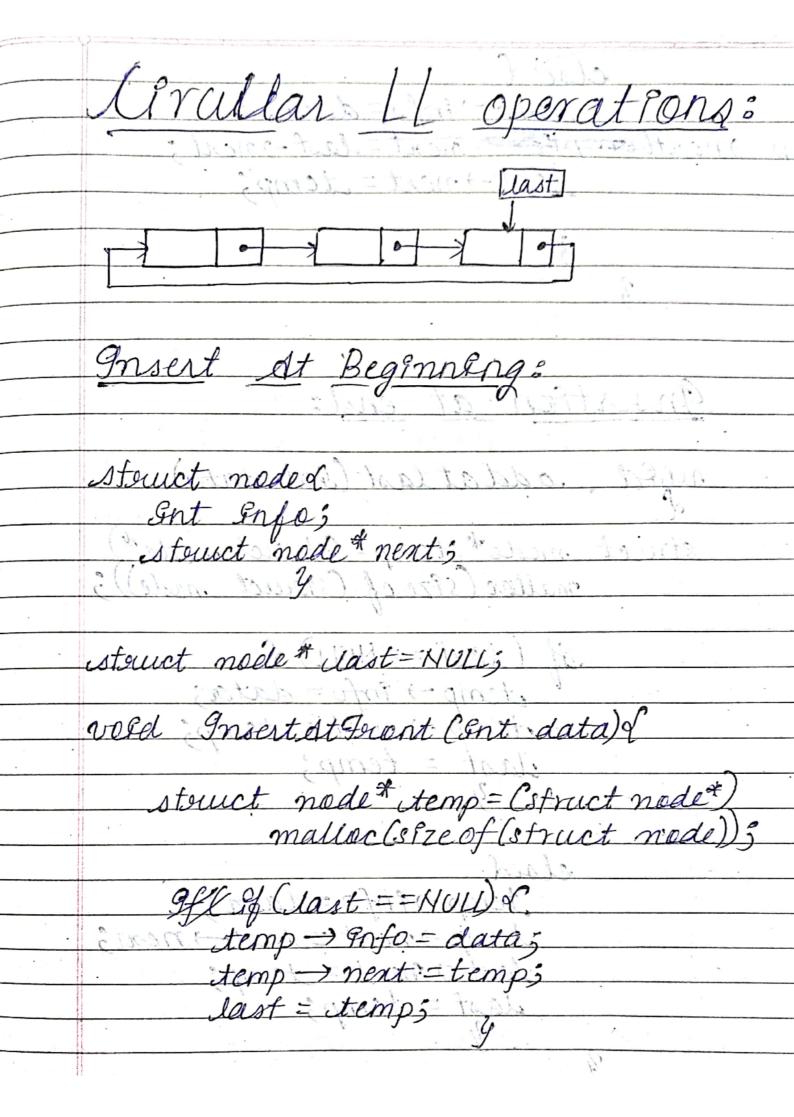
Cercular Janked Lest I descular wenked lest where the last element points to the frest Chead hence forming a circular chain There is no node hosniting to NULL, endreating the alesence cof any end mode. In cercular lenked lests, we have a head pointer leut not starting of the lest. data next data next data next Operations: Operations can be exactly performed Ike sengly denked dest 9th Just that we have to maintain an extora pointer to check If we have gone through the linked list once. Gravessal: It wan be achieved by creating a new stouct Node * pognter p, which start from the head & goes through the est until It points again at the head There There is a head but no starting to this cercular serked best, we have This head posites gust to start or encept en thes lest & for our convenience





```
head = (struct Node *) mallec (size of Cstruct Node)
second=-11-
thesa = - 11
fourth = -11 -
head -> data = 4;
head - nent = second;
second - ) data = 3;
 second -) next = there;
therd - data = 6;
theid next = fourth;
fourth -) data=1;
fourth -) nent= head;
pf ("Rencular Jerked dest lufore inserten (n");
 Anked Let Traversal (head);
head = Insertat First (head, 54);
-11 — (-11 -, 58);
-11 — (-11 -, 59);
                  (-11 -, 58);
Pf("Cercular Lanked Lest after anserteon \n");
lenked Lest Towares al (head);
outurn 0 ;
```

6/p=	Aread - Caternat Vede
Cercular Tenked	Lest before Enserteen
Element & 4	
€lement & 4 €-11-3	Partie
-t1	
† !	hours outer to
	head - Doest = weers
	19st after ensertion
Element & 59	words deta = 33
-11-58	second -) mart - theed
-11-54	
	thending date = 3.
113	thereto may to frote
-116	
<u> </u>	formeth - a determine
	for the man on a fact that



node of new node contains the ref. of the new first nade	
node of new node contains the reg of the previous	Q
TOW! THERE	Ì
else d	7
temp -) anfo = data; temp -) next temp to = next = last -) next;	
temp -) nextlempte next = last -) next;	
last -> nent = temp;	
3	
2	
Garent dt Bennange	
gnsert Pon at end:	
riold add at last (Int data)	
α	
struct mode * temp = Cstruct mede *) mallor (size of (stouct mode));	
malloc (core of (stouct mode));	
marie size of constant	
St (last == NULL) {	
etemp -> enfo = data;	
temp - neat = temps	
uast = temp;	
in (in the second of the secon	
of the same to metal to the transfer of the	j.
elsed	_
temp -) Info = data;	_
temp - nent = last - nent3	
last > nent = temp;	
last = temp;	
3	
3	

	vold Ensertapter ()
	4
	Int data, value;
	atom of death to a death
	struct node * temp, *n;
	of ("In Enter the mas after which
	you want to enter: (n");
	pf ("m Enter the mg. after which you want to enter: \n"); 8 canf ("%d", & value);
-	demp = last - next;
_	1.00
	dod Sf (temp-) Gnfo == value) of
	Je (1011) - 911/0 == value/9
	m = (struct node*) malloc(size of (struct
	m = (stouct node*) malloc(size of (struct node));
	pf ("/m Enter data to le gnserted: "); scanf ("/d", & data);
_	scanf ("%d", & data),
	m > Co Co = double?
()	$n \rightarrow 9nfo = data_3$ $(n \rightarrow next = temp \rightarrow next_3$
	temp -) nent = n;
	borrow i.
	Sf (temp == last) of
	last=1)
ur.	break;
	4

	else &
	temp = temp - nent:
	temp= temp -) ment;
	y robble (temp!= last -> next);
,	Thesits
	3. Contract to another terroria.
	Charles are a second
	Langton to delate in its
	Function to delete à particular
	element
,	Merrica (Maria) Harris
	road delete Wode (struct node & head) ant key)
	Tota alletevode struct node thead
	ent key)
	9 1000
	If (head == NULL) seturn;
7	seturn;
	as Cateuret sicile Danaller (specific
-	stouct mode * curs = head, * prev;
1	while (curs data != key)
	of the state of th
-	If (cur -) next == head)
-	a strain of the color
-	pf ("Green node not found-\n");
Continue (major of the continue of the continu	conner menterna
A STATE OF COLUMN	break;
And the state of t	Part == and 140
Alle appropriate the second	y restant
A CONTRACTOR OF THE PERSON NAMED IN COLUMN NAM	prev= ceus;
The state of the s	curs = curs - nent;
Service Servic	3
and the state of	

11 Check If node Is only node If (run -) nent3 == head) head == NULL; free (curi); veturn; If more than one node, check of It's first mode prev= head; whole (prev-next!=head)d prev= prev-nent; head = curs - nent; prev -> nent = heads. free (curs); 11 check of clast node -else of Cours - next = = head && curi==head) prev-neat = head;

free(ceur); prev -) next = curs -> next; free Cours);