

THE Y		
	9)	WAP Implement doubly list with
/		primitive operations:
	a)	Create a doubly linked list
	62	Insert a new node to the left of the node
	c)	Delete the node based on a specific node
	d	Display the contents of the list
	(	eranic de la
		#include < stdio-h>
		#include < stdlib:h>
		in the standard and the
		struct node?
		int data;
		strict node *next;
		struct node * prev;
		J; la resultation de la result
		struct node thead = NULL;
		ein and the company of the company
		iroid insert-beg()
		<u>{</u>
		struct node *new_node;
	-	nen nøde = (struct node*) malloc (size of
***		(struct node);
		printf ("Enter the item n');
	-	scanf ("/.d", frent node -7. data);
25 6	5.	ner-noole=>next=NULL;
	4,1	ners-nede->-prev =NUKL;
The second	-	
		if (head == NUKL) {
		head = next-riode;
14/2	ta j	
		else 1 nen næde -7 neset = head;
		head-zfrev = new noole
-	-	head-sprew node;
	$-\parallel$	1
		2 )



word insert end () { struct node \* new node, \* temps; new node = (struct node\*) malloc (vigey (struct node)); printf ("Enter the item)"); scant ("/.d", frest node->data); new mode-Treat = NULL; nen node-> prev =NULLi if (head=NULL)? head = new node; else temp = head; while (temp->next!=NULL) temp = temp = next; temp->next=new-node; new node -> previtemp word insert left () { int listele: struct node \*now node, \* temp; printf ("Enter the element in the list (n''); scanf ("/-d", flistele); nent nede = (etruct node \*) molloc (size of (struct node)): frientf ( 'Enter the new node datal') scant ("/.d", Iner node-7 dates) new-noole->next=NULL; new node->prev=NUCL; if (head=-NUCL)

printf (" tompty list | n'); temp = head; while (temp->next->data!=listele){ temp=temp=next; if (temp = = NULL) & printf ("Element is not found in new-node->next=temp->next; temp->neset=new-node; rent node - sprev = temp; new\_node -> next -> prev = new\_node: word del () { street node \* temps if (head == NULL) { grintle ("Empty list \n"); Reliern; frint ("Enter the element to be olcleted m'); scanf ("/-d", fele): taln): temp = nead; while (temp -> plata!-ele) { tem: temp-next; if (temp==NOKK) { prints ("Element is not in the list \n'); 2 break;



```
if (temp == head) {
      head = head = Inent;
   che if (temp->next==NULL) {
        temp - temp -> free;
     temp-7/200-7 next = temp->next
void display () ?
   struct node * temps;
   while (temp!=NULL) {
      printf ("/.d )t", temp-rdata )
     temp = temp-> next;
   print ("(n");
```