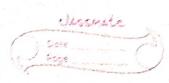
	Date
	Page
1-2-24	Week 71
	WAP to implement doubly like list with primitive
	operations
	#include < stdio. h>
	#include < stdlib.h>
	stemt Node
	110-10-01-05-01
190	int data;
	sterrt Node * peer;
	struct Node * next;
	};
	steurt Node *ceate Node (int data)
	struct Node * new Node = (struct Node *) malloc (size of (struct M
* All Stellands	if (new Node == NULL)
	printf ("Memory allocation failed \n");
	exit (1);
	newNode → data = data
	new Node -> prev = NVLL;
	new Node -+ pext = NULL;
	return new Node;
	3
	void issert Node (struct Node *head, struct Node *forget, int data)
	{
	strut Nede *newNode = createNode (data);
	if (forget -> prev! = NULL)
	- { · · · · · · · · · · · · · · · · · ·
	forget → prev → next = new Node:
	new Node -> prev = forget -> prev ;
	]
	else
	head = new Node;

newNorde -- next = forget; forget -- prev = newNorde;



```
void delete Node (eteut Node * head, int value)
      strut Node * weset = head;
     while (current != NULL)
          it (wesert - doto == value)
              if (current - prev ! = NULL)
                    werent - peur - next = current - next;
                 current - next - prev = current - prev;
    printf ( node with value "d not found In", value);
void display (street Nocle * head)
     printf (" doubly liked list: 10");
     while (head ! = NULL)
          printf (" /d e> " head - data);
    printf ("NULL \n");
```

The second secon	void main ()
https://www.tapicitalisticiaes.org/diseastingscontages/diseast	1 1 1 * hand = UVLL
minute de la completa esta productiva e	head = create Node (1);
	head -> next = create Node (2);
	head - next - prev = head;
	head - rest - pseu + sade (3);
	head -> next -> next = cerote Node (3);
	head - next - next - prev = head - next;
	dieplay (head)
	insert Node (head, head - next, 10);
add y a company or that it was provided to the contract of the	peintf ("ofter insation: \n");
	display (head)
	The state of the s
	delete Node (head, 2);
	printf (" after deletion: \n");
	dieplay (head);
	3
	Output - /
	doubly liked list:
	1 \ightarrow 2 \ightarrow 3 \ightarrow NULL
	after incertion:
	doubly liked list:
	1 → 10 ↔ 2 ↔ 3 ↔ NULL
and the same of th	after deletion:
	doubly linked list:
	1 ← 10 ← 3 ← NULL
	The second of th

k" let