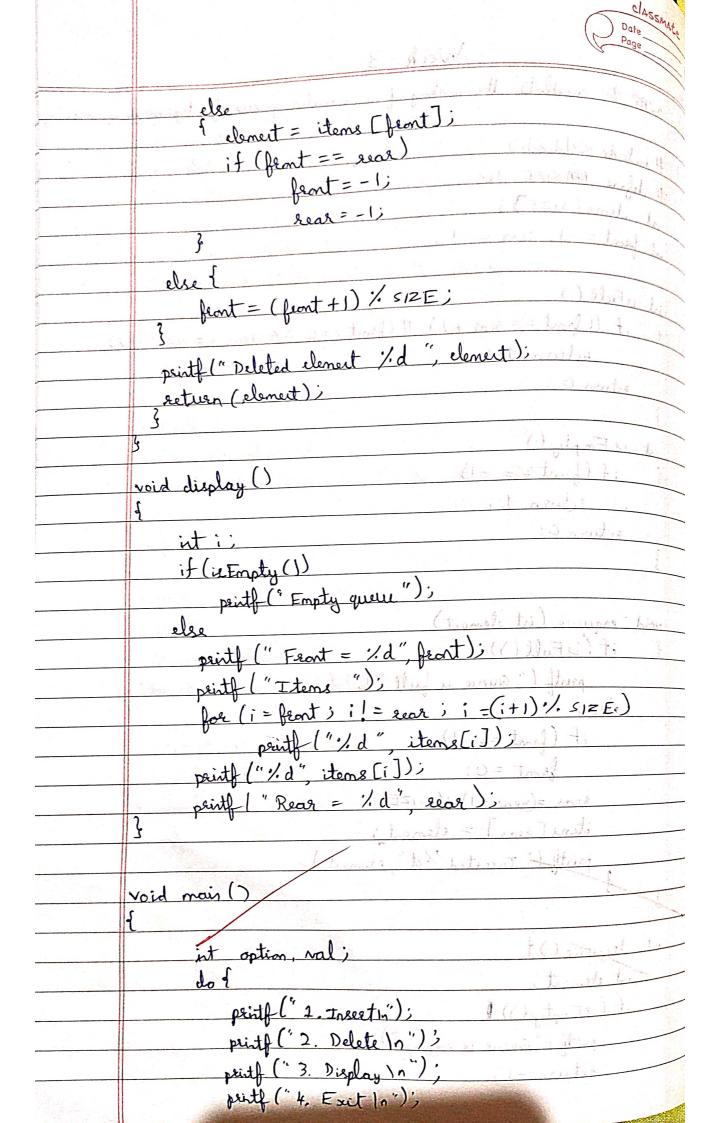
Week-3 11-1-24 WAP to similate the working of a circular queue of integers using an array (3b) #include < stdiah) # define MASIZE 10 int items [SIZE]; int flont = -1, sear = -1; int in Full () { if ((front = = rear +1) || (front = = 0 & & rear = ± SIZE -1)) return O; int is Empty () if(feant = = -1)return 1; return 0; void erqueue (int element) if (is Full ()) if trail "bis = to printf (" Queue is full "); if (front == -1) not sear = (sear +1) :/ 512E; items [sear] = element; printf (Inserted "d" element); int dequerie () { it element; if (is Empty ())

paintly (" Queue is empty ");

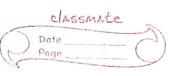
return -1;



	printf ("Enter your	option: In"),	
	seaf ("/d", So	stignitude of the last	
	switch (option)	and the last place	
	5	V V	
	case 1:	rqueue (); () the	
		k; Shallbergham	
		dequeue (1; stant toward	
	if (n	al (= -1)	
	ρε	itf ("the number deleted is: "d", val);	
	beeak :	too to bow touch	
	costs, cospilar		
	2 seak 0		
	I while (option 1 = 4)	joh "t about Lucto I team work!	
	}		
à	I for it I wall truste - (show touch +) and truste		
	Output:	sa stransat	
	I. Insert	1. Insect atal	
	2. Delete	2. Pelete mans	
	3. Display	3. Dieplay	
	4. Exit	olle Exit: handin	
	Exter your option: 1	Exter your option: 2	
	erter the element: 1	Peleted element > \$ 1	
	Inserted -> 1	the number deleted is: 1	
	1. Insert	tad 1 Insect	
	2 Delete	(2. Delete) with	
	3. Display	3. Display	
		4, Exit	
_		Enter your option: 3	
	exter the element: 2		
	Frslited → 2	BItens → 2	
		Real → 1	



ha di agai ana agai an saon i ka		
	1 1 insert	t and displa.
(Q)	WAP to show implementation of	we have been
	WAP to show implementation of insert singly listed list.	
	1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
	H: 1 de/ stallib.h	1 C Shrin
	4 1 2 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3 1 3	
lear.	1 2 x x L (1 th b) where 1 1 x	
	3.	2000
	void insert (stant Node ** head, int o	lata)
		£ \ 00 (-, A.
	stant Node *neumode = (stant Node	maller (single)
	stant Node *noumode = (stant Node?	maller (single)
	newnode ->	in the state of
	newnode → data = data;	tonatta (single)
	newrode -> data = data: newrode ->	Paret 1
	newnode → data = data;	1 2020 T 2 2101,0 S
	newrode -> data = data: newrode -> next = *head:	topot 2 stolar s godge: a s
	newrode -> dota = data; newrode -> next = *head; *head = newrode; }	topot 2 stolar s godge: a s
	newrode -> dota = data; newrode -> rext = *head; *head = newrode; }	tonat i stolar s godge: a s type may sta
₹	newrode -> dota = data; newrode -> next = *head; *head = newrode; }	tonat i stoha s godge: a s liet s
7	newrode -> data = data; newrode -> next = *head; *head = newrode; } void display (struct Node * rode) {	tonat i stolar s godge: a s type may sta
	newrode -> data = data; newrode -> next = *head; *head = newrode; void dieplay (struct Node * rode) printf (n Linker list = ");	tonat i stolar s godge: a s type may sta
	newrode -> data = data; newrode -> next = *head; *head = newrode; } void display (struct Node * node) printf ("n Linkerd list = "); while (node != NULL)	tonat i stolar s godge: a s type may sta
	newrode data = data: newrode next = *head: *head = newrode: void display (struct Node * node) printf((node != NULL) find (node != NULL)	tonat i stolar s godge: a s type may sta
	newrode -> data = data; newrode -> newrode> newrod	tonat i stolar s godge: a s type may sta
	newrode > data = data; newrode > next = *head; *head = newrode; void display (struct Node * node) printf ("/d" inde > data); node = node > pest; rode = node > pest;	trace to the start of the start
	newrode -> data = data; newrode -> newrode> newrod	tonat i stolar s godge: a s type may sta



the state of the s	
	void main ()
	£
	steut Node *head = NULL;
	insert (& head, 100);
	issert (2 head, 80);
	insert (& head, 60);
	insert (& head, 401;
	insert (4 head, 20);
	display (head);
	3
	Output
	Linked list: 20 40 60 80 100
	De.
,	