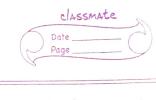


using linked representation. // stack implementation #include < stdio-h7 # include < stollib-h > word puch (); word frogs (); void display (): struct node (int data; street node *next; struct node *top: NULL; int main () { int choice; charch; do 1 print ("\n1-Push \n2-Display \n3. Pop\n");

printf ("In Enter your choice;"); scanf ("-/-d", & choice); switch (choice) { case 1: push (); break (); care 2: chisplay (); break;

classmate
Date
case 3: 120ps();
break;
3
printf ("In Do you want to continue
(y11y):");
folish (stdin);
scant ("/·c", fch); 3 while (ch == 'y' ch == 'y');
3 while (ch == 'y' ch == 'Y')
void push () 2
int item;
struct node *newnode;
printf ("Enter the element \n");
scarf ("/od", fitem);
nennode = (struct rude *) mallix
(size of (struct node));
newnode -> data = item;
newnode-Inext = NULL;
if (top==NOLL)
top=newnode;
else
newnode-Inext = top;
top = neumode;
P
void pop () {
if (top == NUCC)
printf ("stack is empty");
else l

classmate
Date
printf ("Element removed is "/-d:")
toli -7 Malas
top: top-7 next;
2
void display () {
strict node * temp;
temp=top;
if (top=NULL)
frentf ("Stack is empty");
while (temp! = NULL) {
printh (" of d , temp
temp = temp = next;
2
// queue implementation.
#include = stdio-h>
#include < stdlib.h>
struct node ?
struct node *next;
<i>}</i> ;
ereid insert ();
world display();
void del ();



struct node * rear=NULL, * front=NULC; int main () { ent choice? char ch = 'Y'; prints ("In guere implementation using linked list(n'); printf (" |n1. Create In2. Display n3. Delete In4. Exit In") prints ("In Enter your choice:"); scanf (" /. d", Schoice); switch (choice) ? case 1: insert (); break; case 2: display (); break; case 3: del(); break; case 4: ch = 'n';

3 while (ch=='y'|1|ch=='y');
word insert() {

struct node * newnode;

newnode = (struct node *) malloc (size of (struct node));

printf ("Enter the element: \n");

scanf ("'/.d", f newnode-7data),

newnode -> next = NULL;

		classmate	
,		Date	
.p=====			
	if Crear == NUCL) {		
	rear = nennode 1		
	front - nemode;		
J	else E		
	rear-Inext=newnode:		
	rar=newnode;		
٠	3		
	word del () {		
	2 1	,	
	if (front == NULL) ? printf ("Greece it compty	(1n');	
	reteisn;		
J	else É		
	printh ("Deleted ele is -/	d 1)	
	brent-data)		
	if (front == rear) {		
	printf (" Greece is con	ptylon')	
	front = NULL;		
	reas=NULL;		
	3		
	else 1		
	front = front -> next;		
	7		
	eroid display () {		
	struct node atemp;		
	the (front == NUCL) f		
	print (" Prieue is emp	ty (n');	
	,		

classmate return; temp= front; unite (temp!=NULU) { print (" of d", temp- s data); temp: temp > next;