	1 1
2.	Stack implementation using linked list
	Hinclude < stdip.h>
	void push ();
	void pop ();
	void diplay();
	struct node
	(
	int data;
	struct node * next;
	3
	struct node *head = NULL;
	Opacia - Noce
-	void main()
	f int ch
	printf("Enter: 1. Push D. Pop 3. Display:");
	scant (" % d", if (h);
	while (ch! = 4)
	Surve (crit = 4)
	{
	Switch (ch)
	{ Switch (ch) }
	{ Switch (ch) { case 1: push();
	Switch (ch) { case 1: push(); break;
200	Switch (ch) Case 1: push(); break; case 2: pop();
\	Switch (ch) Case 1: push(); break; case 2: pop(1; break;
	Switch (ch) Case 1: push(); break; case 2: pop(1; break;
\	Switch (ch) Case 1: push(); break; case 2: pop();
	Switch (ch) Case 1: push(); break; case 2: pop(1; break;
	Switch (ch) Case 1: push(): break; case 2: pop(): break; (ase 3: display(): break; 3
	Switch (ch) Case 1: push(): break; case 2: pop(1: break;

```
void push ()
     print ("Enter no of nodes:");
scanf ("".d", fn);
     int data, h;
     for (ind iso size it +1)
           struct node "last = head;
           struct hade * new-hade
            = (struct node*) malloc
             (Size of (struct node));
          print! ("Enter the data:");
            scart ("1.d", from node -> data)
            new - node -> next = NOLL
           while (last -) next != NULL)
              last = last -) next;
            last - next = new-node;
void pop ()
    struct node * pto; .
    struct node * ptr1;
    if (head -) hext == NULL)
     [ free (head);
         head = NULL;
      print+ (" Flement popped").
     pto = head;
```

ptr1 = head;
while (ptx -) next! = NULL)
5
 ph1 = ptr;
ptr:ptr -> next;
ptr1 -> next = NULL
free (ptr):
7166 (part 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 print! (" Flement peopled successfully");
 , J
4
or sit is and a sit
uid display ()
Void display ()
 2
 struct node * p = head;
printf ("Stack: \n");
idile(p) = NULL)
\
print (" " /d -> ", p -> data);
p = p -> next;
<u> </u>
printf ("NULL \n");
3

	DUTPUT:
	1. Push
	2. Pop
	3. Display
	4. Exit
	Enter choice: 1
	Enter the data: 5
	Element pushed successfully
	Enter thoice: 41
<u>l'</u>	Enter the data: 4
	Element to pushed successfully
	Enter Choice: 1
	Enter the data: 3
	Element pushed successfully
	Enter choice: 3
ــــــــــــــــــــــــــــــــــــــ	List:
1	5->4->3 → NULL
	Enter choice: 2
	Element popped succenfully
	Enter choice: 3
	List
	5-14 - NULL
	Enter chaice: 4
-	Exited
-	
-	

3.	Queue Implementation using linked list
	Hindude <stdio.h></stdio.h>
	void enquene();
	void dequence ();
	struct node
	<u> </u>
	jet data;
	struct node * next;
	3:
-/	Struct node *head = NUL;
يرسوك	Du.
- 2	(Void main!)
828	1
	printf(" 1. Enqueue J. Degneue S. Display 4. Exit");
	int ch;
	while (ch!=4)
	print ("Enter choice: ");
	scant (""(d", fch);
	switch (ch)
	Swain (W)
	case 1: enquence();
	break;
	care D: dequene ();
	break;
	Case 3:
	display();
	break;
	3
	3
	<i>S</i>

void enqueur () struct node *last = head; struct node + new-node; now node = (struct node *) mallac (size of (struct node)); printf ("Enter the data:"); is scanf (" " d d", & data); new-node -> data = data: new-node > next = NULL; if (head == NULL) head = new node; die [while (last -) next != NULL) last = last ->next;] printf ("Node added succenfully \n"), void dequeue() struct node * pty; if (head == NULL). printf ("List is empty (""); S ptr = head; head = ptr -> next;
free(ptr);
print + (" Node deleted from beginning).

	void display ()
	2
	struct node + p = head;
	print+("\n");
	while (p! = NULL)
	The second secon
	print (""/.d -> ", p -> data);
	p=p->next;
	3
	printH"NULL(n")
	3
	ENTA:
_	
-57	1. En queue
	2. Dequene
	3. Display
	Enter choice: 1
	Enter data: 1
	Enter choice: 1
	Enter data: 0
	Enter choice: I
	Enter data: 3
	Enter choice: 3
_	test Queux:
_	1->2->3+ NULL
	Enter choice: 2
	Node deleted from beginning
	Enter choice: 3
	Queue:
	D-3 -> NULL
	Enter choice: 4
	Exited
	II