	PAGE NO. DATE / /
	DATE 1 1
	Data Structures - Lab4
1.	Circular Queue
	#include < stdio. h>
	# include < stdlib. h>
	# include < process it >
	# define que_size 3
	int item, front =0, rear =-1, q, [que_size], count =0;
	Void insert rear ()
	S () I c) + + 1 p
	if (count == que_size)
	printy ("que ne overflow");
	return;
	4004 - (Year +1) 1/2 au sine 2
	rear = (rear +1) / que_size; q [rear] = item;
	Count ++; i Hand of Hand
	2
	int delite front ()
	£ 1000 1 1 1
	if (count ==0)
	return -1;
	item = 9 [front];
	front = (front +1) 1. que size;
	Count = count -1;
	retwen item:
	9 (++118+21-1-11-11)
	void displaya, ()
	4
	inti, t;
	if (count ==0)

	PAGE NO. DATE / /
	•
	}
	Printy (" queue is empty");
	return;
1	la de la companya de
	= front;
	printy (" contents of queue \");
	for (i=0; i < count; itt)
	ş 1
	printy (" 7.d x", q [f]);
	(= (++1) 1/1 que_size)
	300
	1 to -1
	Void main ()
	\\
	int choice;
	Joy (;;)
	2 D
	prints (" \n 1. Insect rear \n2. delete pont \n 3. Display
	in a exit in");
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	Scourf (" "/ d", & choice); Switch (choice)
	Switch (choice)
	Case 1: printy (" enter the item to be inscribed: "); Scary (" /.d", & item); insert reas ();
	Scarl (" 1.d", & item); insert reas ();
	break;
	Case 2: item = delcte front ();
	(:tem = = -1)
	month (" queue is empty \?):
	else prints (" item dileted is / d \"; item);
	break; case 3: display q();
	break; case 3: display q (); break; default: exit(0); 33 getch(1; 3
	actch (1: 4

	PAGE NO.		
	DATE		
Double ended Queue			
#include < stdio.h>			
# include <stdlib.h></stdlib.h>			
# include < process. h>			
# define que_size 5			
int= 0, 1 == 1, Ch;			
int j= 0,1==1, Ch; - int item, q [10];		-	-
int is full ()			
return $(r = = q \operatorname{Size} -1)^{\frac{1}{2}} 1:0$,)		
3		-	
int is empty ()			
\$			
return (} > 8) ? 1:0;			
) and a sud amount ()			
Void insest rear ()			
if (is full ())			
f Cis fatte cr			
prints ("queue overflow \n");			
yetusu;			
7		-	
Y= Y+1;			
q[r]=item; }			
Void delete front ()			
\$ 0	•		
if (is empty ())			
& printy (" queue is empty \n");			
retuen 9		71	
printy ("item deleted is /d \")" [(1)++	1);	
if (1>x)	V		
, ,			

	PAGE NO. DATE / /
	DATE / /
	\$ 1 =0; Y=-1;
	void insul-pout ()
	$\begin{cases} 1 & \text{if } (1 = 0) \\ 1 & \text{if } (1 = 0) \end{cases}$ $\begin{cases} 1 & \text{otherwise} \\ 1 & \text{otherwise} \end{cases}$
	D & 0 = 1 - 1-
	aff = item;
	retuen;
	the state of the s
	else if $(1 = = 0) k \cdot (Y = = -1)$
	$\xi q [1+(r)] = item;$
	retuen
	3
	else print ("Insection not possible \");
	- 3
	void delete rear []
	1 () () () () () () () ()
	f printy (" queue is empty \");
	if (is empty ()) it is empty (); return; 3 return; 3
	printy 1" item deleted is 1/2 d \", 9/[(x)1]);
	il (+ > x)
	$\frac{1}{2} = 0; y = -1; \frac{3}{2}$
	Void display ()
	3 inter
	if (is empty U)
	& prints (queue empty 10);
	retuen; 3
	for (i= ; i <= 7; i+t)
	prints (" %d \n ", g, [i]);
	3
	Void main ()
	\$
- 11	,

	PAGE NO. DATE / /
	for (;;)
	1 A A A A A A A A A A A A A A A A A A A
	frints ("1. insert rear \n 2. insert front \n 3. delete rear \n 1. delete front \n 5 display \n 6.exit
	dulete real in 4- debete front 115 desputy result
	The state of the s
	Scanj (" /.d", k.ch);
	Switch (ch)
	() to the the dead of ").
	(asel: prints ("enter the item \n"); Scary ("1.d", & item);
	insest real ();
	break;
	Case 2: prints (" cutes the item \n");
	Case 2: printz (" cutes the item \n"); Scary (" / d', & item);
	Inscet fout ();
	break?
	Case 3: délete rear l';
1 1	breaked none land Ind Just
	Case 1: delete pont 1);
	break)
	Case 5: display ()
	break;
	Gefault: (XIT (D)
,	