```
#include <stdio.h>
#include <stdlib.h>
struct Node {
   int data;
    struct Node* next;
};
struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*) malloc(sizeof(struct Node));
   newNode->data = data;
    newNode->next = NULL;
   return newNode;
void push(struct Node** top, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = *top;
    *top = newNode;
int pop(struct Node** top) {
    if (*top == NULL) {
        printf("Stack underflow");
        return -1;
    } else {
        struct Node* temp = *top;
        *top = temp->next;
        int popped = temp->data;
        free(temp);
        return popped;
int peek(struct Node* top) {
   if (top == NULL) {
        printf("Stack is empty");
        return -1;
    } else {
        return top->data;
void dispStack(struct Node* top) {
```

```
if (top == NULL) {
        printf("Stack is empty");
    } else {
        printf("Stack elements: ");
        struct Node* temp = top;
        while (temp != NULL) {
            printf("%d ", temp->data);
            temp = temp->next;
       printf("\n");
void enqueue(struct Node** f, struct Node** r, int data) {
    struct Node* newNode = createNode(data);
    if (*r == NULL) {
        *f = *r = newNode;
        return;
    (*r)->next = newNode;
    *r = newNode;
int dequeue(struct Node** f) {
    if (*f == NULL) {
       printf("Underflow");
        return -1;
    } else {
        struct Node* temp = *f;
        int dequeued = temp->data;
        *f = temp->next;
        free(temp);
        return dequeued;
void dispQueue(struct Node* f) {
   if (f == NULL) {
        printf("Queue is Empty!!");
    } else {
        printf("Queue elements: ");
        struct Node* temp = f;
        while (temp != NULL) {
            printf("%d ", temp->data);
           temp = temp->next;
```

```
printf("\n");
int main() {
    struct Node* stack1 = NULL;
    struct Node *q1 = NULL, *q2 = NULL;
    int c, d;
    printf("Enter your choice:\n1. Push to Stack\n2. Pop from Stack\n3. Peek
Stack\n4. Display Stack\n5. Enqueue\n6. Dequeue\n7. Display Queue\n8. Exit\n");
    while (1) {
        printf("Enter your choice: ");
        scanf("%d", &c);
        switch (c) {
            case 1:
                printf("Enter data to push: ");
                scanf("%d", &d);
                push(&stack1, d);
                break;
            case 2:
                d = pop(&stack1);
                printf("Popped element is: %d\n", d);
                break;
            case 3:
                d = peek(stack1);
                printf("Top element: %d\n", d);
                break;
            case 4:
                dispStack(stack1);
                break;
            case 5:
                printf("Enter data to enqueue: ");
                scanf("%d", &d);
                enqueue(&q1, &q2, d);
                break;
            case 6:
                d = dequeue(&q1);
                printf("Element %d was dequeued\n", d);
                break;
            case 7:
                dispQueue(q1);
                break;
            case 8:
```

```
return 0;
    default:
        printf("Invalid input\n");
    }
}
return 0;
}
```

## Output:

```
djprweu5.nx0 --dbgExe=C:\\msys64\\ucrt64\\bin\\gdb.exe --interpreter=mi
Enter your choice:
1. Push to Stack
2. Pop from Stack
3. Peek Stack
4. Display Stack
5. Enqueue
6. Dequeue
7. Display Queue
8. Exit
Enter your choice: 1
Enter data to push: 10
Enter your choice: 1
Enter data to push: 5
Enter your choice: 1
Enter data to push: 6
Enter your choice: 4
Stack elements: 6 5 10
Enter your choice: 3
Top element: 6
Enter your choice: 5
Enter data to enqueue: 12
Enter your choice: 2
Popped element is: 6
Enter your choice: 5
Enter data to enqueue: 2
Enter your choice: 5
Enter data to enqueue: 6
Enter your choice: 7
Queue elements: 12 2 6
Enter your choice: 6
Element 12 was dequeued
Enter your choice: 7
Queue elements: 2 6
Enter your choice:
```

Shurt Nocle  shurt Nocle  int dab;  shurt Nocle' next;  shurt Nocle' next;  shurt Nocle' next;  shurt Nocle' newnode - (thurt Nocle) mallor (size of (s knet nocl))  new Nocle - data; data;  new Nocle - next = NULL;  Shurt new Nocle;  3  Void frush (shurt Nocle + top, wit data)	The same of the sa	
Shuet Nocle  2  int dab;  shuet Nocle' newl;  3;  Shuet Nocle' newl;  \$\frac{1}{2}\$  Shuet Nocle' newl;  \$\frac{1}{2}\$  Shuet Nocle' newnode - \( \text{Stuet Nocle'} \) malloc (\( \text{size of (S \text{fuet nod)} \) \\  \text{New Nocle + new node - \( \text{Stuet Nocle'} \) malloc (\( \text{size of (S \text{fuet nod)} \) \\  \text{New Nocle + new node : \( \text{Surf new Nocle : } \)  \text{New Nocle + new node : \( \text{Surf new Nocle : } \)  \text{Shuen new Nocle : \( \text{Surf new Nocle : } \)		
Struct Nocle  2  int dato;  struct Nocle' nert;  3;  Struct Nocle' newhode (wit data)  5  Struct Nocle' newhode - Church Nocle' malloc (size of (struct nocl) new Nocle + new nocle - new nocle + new Nocle;  3  Shuch Nocle of Nocle of Nocle of Nocle of Nocle of (struct nocle)  New Nocle of New Nocle;  3  3  3  3  3  3  4  4  4  4  4  4  4		PAGE:
# unduct (stellis h)  Shuet Node  int date;  shuet Node newl;  shuet Node chok Node (wit date)  shuet Node hewnode - Church Node (size of (struct node)  new Node > data;  new Node > ment = NULL;  Setum new Node;  3  4	66)	
Shuet Node  2  wit dato;  shuet Node* newl;  3;  Shuet Node* Creak Node ( wit data)  2  Shuet Node* newnode - Church Node* malloc ( size of (speet node) new Node + clafa = clata;  new Node + clafa = data;  shur new Node;  3		# unduals Litellis h>
wit dato;  struct Noch " Create Noch ( wit class)  struct Noch " hewnode - (truct Noch ) malloc (size of (struct noch)  new Noch - class = class;  new Noch - next = NULL;  Seturn new Noch;  3;	1	(31001= 301) h
skuet Noch vert;  3;  Skuet Noch vert Chak Noch (wit data)  3  Skuet Noch vert newhoode - (kuet Noch vert malloc (size of (skuet noch) new Noch vert vod)  new Noch vert = NULL;  Sehan new Noch;  3		
Skuet Noch "Creak Noch (wit data)  3;  Skuet Noch " Creak Noch (wit data)  Skuet Noch " newnode - (Kruet Noch) malloc (size of (s fruct noch)  New Noch - clata : clata;  New Noch - next : NVII;  Sehan new Noch;  3	-	
Skuet Noch "Chak Noch (wit data)  Skuet Noch "newnode - (kuet Noch ") malloc (size of (skuet noch)  New Noch -> clata = clata;  New Noch -> next = NVII;  Sehin new Noch;	-	
Skuet Noch " Creak Noch ( wit data)  3  Skuet Noch " newnode - Church Noch" malloc (size of (s fruct noch)  New Noch - claba = claba;  New Noch - next = NVII;  Sehven new Noch;  3	-	
Skuet Noch " Creak Noch ( wit class)  \$  Skuet Noch " newnode - Church Noch " mallor (size of (s pruet noch)  New Noch - class = class ;  New Noch - next = NVII;  Sehin new Noch;  3		
Skuet Noole* newnode - (Kruet Nocle) malloc (size of (spruet nod)  New Noole -> claba : claba;  New Noole -> next = NVLL;  Sehven new Noole;		
Skuet Noole newwoode - Ckuet Nocle malloc (size of (spect wol))  new Noole -> next = NULL;  Sehen new Noole;		
new Noole -> next = NULL;  Sehen new Noole;		
new Noole -> next = NULL;  Sehen new Noole;		Siller North newhode - Greet Noch malloc (size of (s knet wol)
2 Sehan new Node;		Musived 7 clara = clara;
3 Engling a Anti-Althord		
A STEWN OF A THE A STANKE		retiren new Nocle;
void fush ( skuck Node ++ top, int data)		Today a Lot of third
old fush I shick Nocle ++ top, iid data)	16	il to the to
	3	Figh ( SMick Node ## top, wit date)
chief a ch	7	Charles Charles
Street Node * new Node = create Nocle (data);		Stulet Noch " New Noch = create Noche (dota);
new Nocle -> nent = # tof		NIWNOCLE -> Nent = # tof
* top = newNode;		100 = New Node;
4 Atrit	4	
Carling to the part of the par	110	( Soletie dural ( "At ) Hours
in pot wheet Nocle * + bel)	Lat	pop ( street Nocle ** top)
	3	
if ( * /06 == NULL)		y ( * /ob == NULL)
& friend (" Slack underflow")		& fruit (" Stack under (0,2")
Kentan		- Actual Comment of the Comment of t
else		else
E at land a second a		
Shutet Noch " temp = # lop;		Stillet Node temp = # top;
top = temp > next;		top = temp > next;
test int possed = temp -> clasa:		The cut posped = temp -> data.
Top = temp > neut;		1 op = temp > newt
g g Relectors)	1 9	3 School (temp)
popped;		popped;

		PAGE:
	E if (top == NULL)	minist diagram ton
	il (La man)	tillete Andrew W. I.
	\$ (10% = = NOLL)	1
	S when the	(4).
		dada Tas
	else	one lawn,
	900.00 1-0 11	2.7
	Schuln top > data;	Street Associate
	void dispersack (skuet Node " top)	Booth barries
	if ( hop == NULL)	S Devinar
		18 1 PSN 28
	printf (" Stack is empty"); selvern ?	2
		1
	else.	1 ) ARING BOOM
	frint (" Stack Clements: ");	2 4 4 4
	Shuet Node " temp = top; while (temp! = NULL)	NA 101017
	ashile (fump! = NVLL)	marianiii
	f. TICH	Tere 2.1
	funt ("1-d", temp > data); funt = temp > ment;	S. S.
	3 temp = remp > ment;	12 ) Bad 300
-	Ment ("In")	3
		43(+)15
	off C. State unduling ?	and g
	18337	STR
	Alle a .	3)30
	And the state of t	
	aliaba de la desta de la constante de la const	*
	have bother in to	
	The state of the s	
YE	Control of the second	
4000		August 1

1		DATE
1	switch (c)	12214
1	3	t P.A. to State
1	case 1:	3,612 35 0
1	frent (" Fishe o	(ala to kush")
	scarf (" 1 d	late to kuth"), ", &d);
-	kush C &stack	t, d);
1	bleak;	C Steller
1	0-11 0	- Person Court
-	Cast 2:	130 3
	a pop (xsia	ekt);
		ekt); d element i 1.dh", d);
	break;	Control Control
	Call 3:	To also says
	d = peck (sta	ekt);
	print (" top e	lement: -/-d4\u", d);
	heak;	3 10003
	Cast 4:	A Mal
	dupstack (stac	6).
	Cak 5:	
	print/("Cnps	data to enquere");
	Seanf ["-1.d	(1, 2d); (1, 20);
	enquire ( so	N 8 08 1)
_	break;	y 1 x vn, a);
	Can 6:	
	d - downer	Canix
	britt Cu con	(ROJ); rent 1. of was dequired", &d)
	pay 1" fun	vent 1. d was dequered " & d)
	near;	149/
	(mg( ).	
	disp Duene ( of	);
	Pleak;	
-	Cay 8:	
8 9 9	return 0;	
	default: brisy ("Zhva	12 P. 1 (1)

	PAGE DATE
	Oulput:
	Enhr choul:
	1 Rush to Stack
	2. Pof Stack
	3. Peek Stack
	5. display stack
	s. Enquell
	6 Requeur
	7. Righlay Ouill
	8- Gut
	Of = Lab (Rigardy);
dp:	Enha Choice 1
1,	Fuhr datail
	Enhe Cloen 1
	Buhr deta:5
	Ruhr choice 5
	Styck elements: (1
	Bihr Cloice 2
	popped
	olupsiace ( stacko.
	COLUMN
	(1200 0 1 100 0 1000)
	2062.2 1600.37
	(6 18) Bear James
	engine ( sight sext a)
	break