

DS-LAB-5

9.04.24 Linked list stacks.

```
#include <stdio.h>
```

```
#include <stdlib.h>
```

```
struct datemode {
```

```
    int data;
```

```
    struct node *next; }
```

```
* top = NULL;
```

```
Void & push (int value)
```

```
{
```

```
    struct node *newnode = (struct node*)
```

```
        malloc ( sizeof (struct node));
```

```
    newNode->data = value;
```

```
    newNode->next = top;
```

```
    top = newNode;
```

```
    printf ("Successfully Added '%d', Value);
```

```
}
```

```
struct node * temp;
```

```
If (top == NULL)
```

```
{
```

~~```
 printf ("underflow");
```~~

```
}
```

```
else {
```

```
 temp = top;
```

```
; int val = temp->data;
```

```
* top = top->next;
```

```
free (temp);
```

```
temp = NULL;
printf ("%d", val);
}

void display()
{
if (top == NULL)
{
printf (" the stack is empty ");
}
else
{
struct node *ptr = top;
while (ptr != NULL)
{
printf ("%d\n", ptr->data);
ptr = ptr->next;
}
}

void main()
{
int choice, value, boolean = 1;
while (boolean) {
printf ("\n 1.push | 2.pop | 3.display | \n 4.Exit ");
scanf ("%d", &choice);
if (choice == 1) {
}
```

2

switch (choice)

{

case 1:

scanf ("%d", &value);

Push (value);

break;

case 2:

Pop();

break;

case 3:

display();

break;

case 4

boolean = 0;

break;

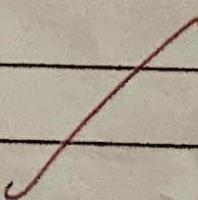
default:

printf ("Invalid output");

}

}

}



Q2

1. Push
2. Pop
3. display
4. Exit

2

successfully added 2

1. Push
2. pop
3. display
4. exit

3

1. Push    2. pop    3. display    4. exit  
3

2  
1. Push    2. pop    3. display    4. exit  
3. deleted.

29.0.17+

linked list queue.

```
#include <stdio.h>
#include <stdlib.h>
struct node{
 int data;
 struct node *next;
};
```

```
struct node *rear = NULL;
struct node *front = NULL
```

```
void enqueue(int val){
```

```
 struct node *newnode = (struct node *)
```

```
 malloc(sizeof(struct node));
```

```
 newnode->data = value;
```

```
 newnode->next = NULL;
```

```
 if (front == 0 & rear == 0)
```

```
{ front = rear = newnode;
```

```
}
```

else {

    rear->next = newnode;

    rear = newnode

void dequeue()

if (front == NULL && rear == NULL)  
{

    printf("empty");  
}

else {

    struct node \*temp;

    temp = front;

    front = front->next;

    free(temp)

    temp = NULL;

}

}

void display()

{

if (front == NULL){

    printf("empty");

}

else

{

    struct node \*ptr = front;

```
while (Ptr != NULL)
```

```
{
```

```
 printf ("%d\n", *Ptr->data);
```

```
 Ptr = Ptr->next;
```

```
}
```

```
}
```

```
void main()
```

```
{
```

```
 int choice, val, bool = 1;
```

```
 while (bool)
```

```
{
```

```
 printf ("1. Enqueue 2. Dequeue 3. Display 4. Exit");
```

```
 scanf ("%d", &choice);
```

```
 switch (choice)
```

```
{
```

```
 case 1:
```

```
 scanf ("%d", &value);
```

```
 enqueue (value);
```

```
 break;
```

```
 case 2:
```

~~```
        dequeue ();
```~~~~```
 break;
```~~

```
 case 3:
```

~~```
        display ();
```~~~~```
 break;
```~~

case 4:

boolean = 0;

break; }

}

}

O/P

1. Enqueue 2. Dequeue 3. display 4. Exit 1

2.

1. enqueue 2. Dequeue 3. display 4. exit 1

3

1. Enqueue 2. Dequeue 3. display 4. exit 2

1. enqueue 2. Dequeue 3. display 4. exit 3

3

open

WIP  
5/10/24