WAP to Implement Single Link List to simulate Stack & Queue Operations.

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

/* Node structure */
struct node {
   int data;
   struct node *next;
};

/* Global pointers */
struct node *top = NULL;     // for Stack
struct node *front = NULL;   // for Queue
struct node *rear = NULL;    // for Queue

/* ---------- STACK OPERATIONS ---------- */
void push(int x) {
   struct node *newnode = (struct node *)malloc(sizeof(struct node));
   newnode->data = x;
   newnode->next = top;
   top = newnode;
   printf("Pushed %d into Stack\n", x);
}

void pop() {
   if (top == NULL) {
      printf("Stack Underflow\n");
      return;
```

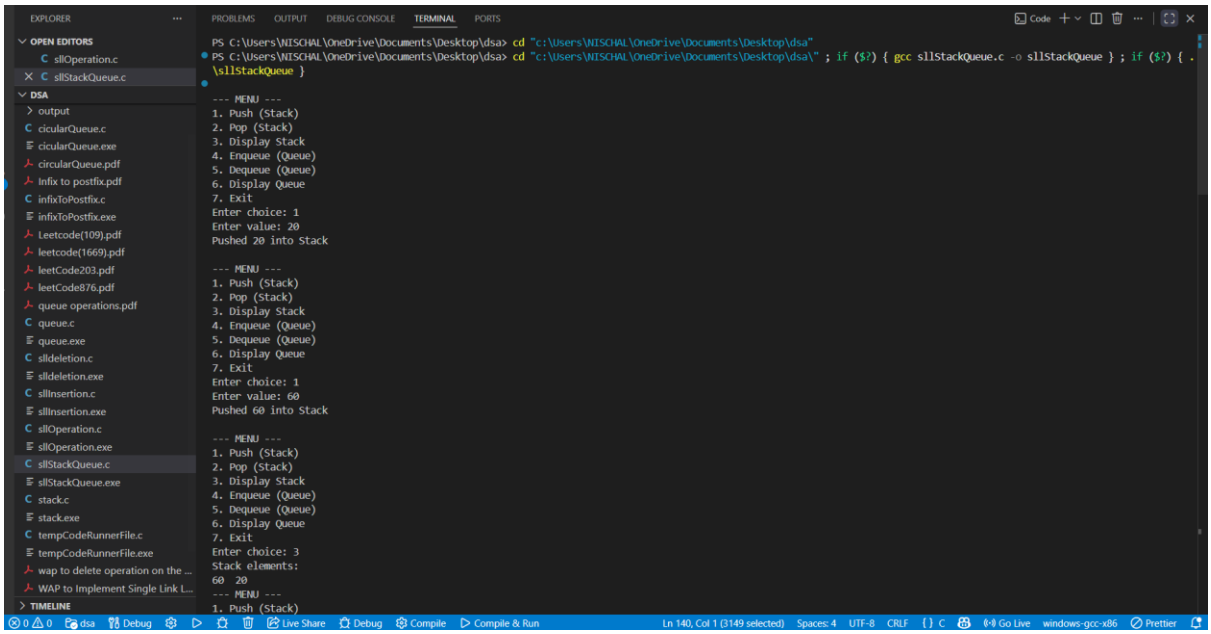
```
    }
    struct node *temp = top;
    printf("Popped %d from Stack\n", temp->data);
    top = top->next;
    free(temp);
}

void displayStack() {
    struct node *temp = top;
    if (temp == NULL) {
        printf("Stack is Empty\n");
        return;
    }
    printf("Stack elements:\n");
    while (temp != NULL) {
        printf("%d  ", temp->data);
        temp = temp->next;
    }

}

/* ---------- QUEUE OPERATIONS ---------- */
void enqueue(int x) {
    struct node *newnode = (struct node *)malloc(sizeof(struct node));
    newnode->data = x;
    newnode->next = NULL;

    if (rear == NULL) {
        front = rear = newnode;
```

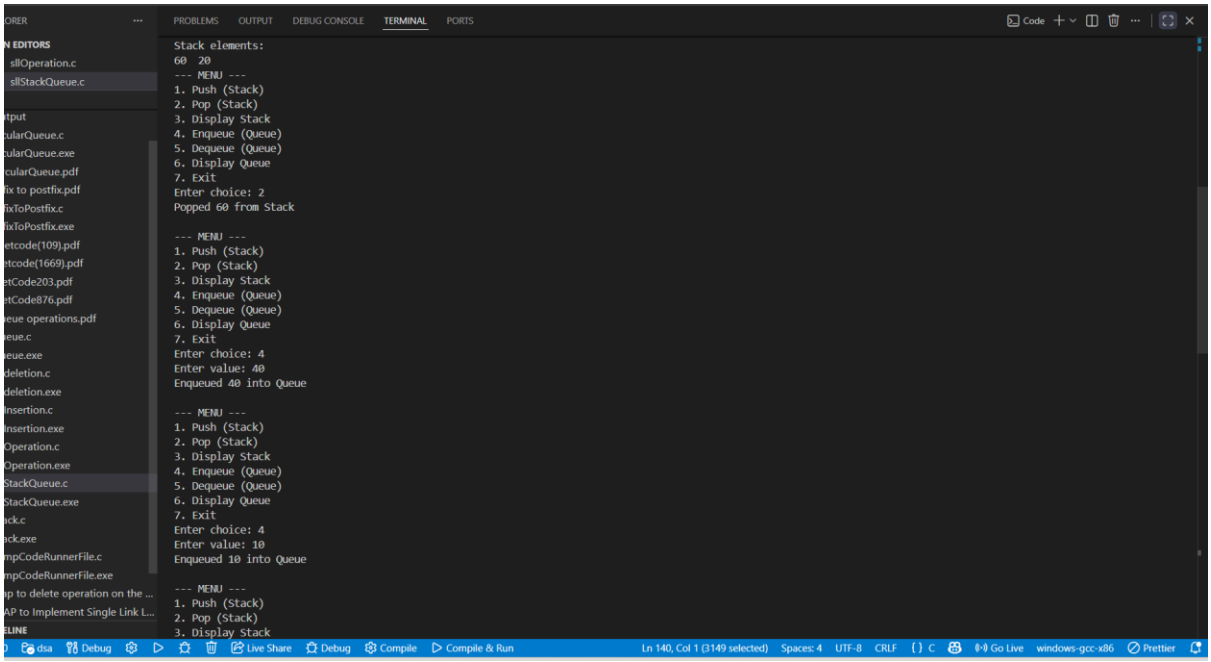
```
    } else {
        rear->next = newnode;
        rear = newnode;
    }
    printf("Enqueued %d into Queue\n", x);
}

void dequeue() {
    if (front == NULL) {
        printf("Queue Underflow\n");
        return;
    }
    struct node *temp = front;
    printf("Dequeued %d from Queue\n", temp->data);
    front = front->next;

    if (front == NULL)
        rear = NULL;

    free(temp);
}

void displayQueue() {
    struct node *temp = front;
    if (temp == NULL) {
        printf("Queue is Empty\n");
        return;
    }
    printf("Queue elements:\n");
```

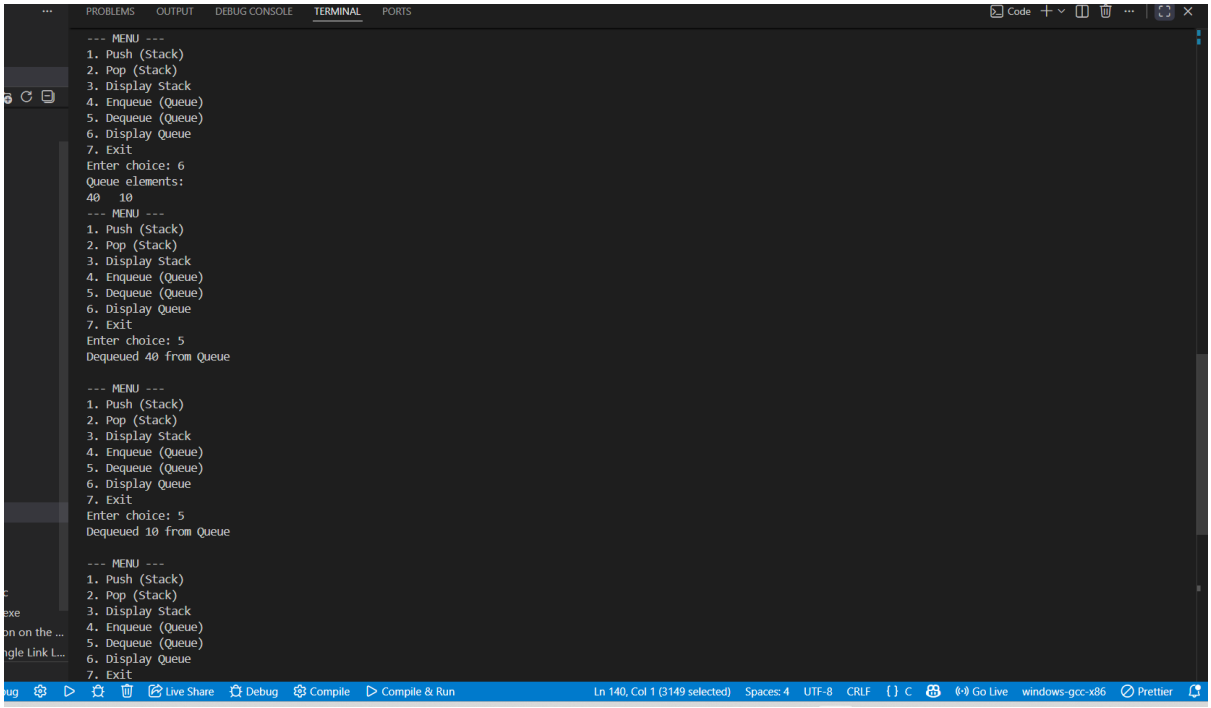
```
    while (temp != NULL) {
        printf("%d   ", temp->data);
        temp = temp->next;
    }

}

/* ---------- MAIN FUNCTION ---------- */
int main() {
    int choice, value;

    while (1) {
        printf("\n--- MENU ---\n");
        printf("1. Push (Stack)\n");
        printf("2. Pop (Stack)\n");
        printf("3. Display Stack\n");
        printf("4. Enqueue (Queue)\n");
        printf("5. Dequeue (Queue)\n");
        printf("6. Display Queue\n");
        printf("7. Exit\n");
        printf("Enter choice: ");
        scanf("%d", &choice);

        switch (choice) {
        case 1:
            printf("Enter value: ");
            scanf("%d", &value);
            push(value);
            break;
```

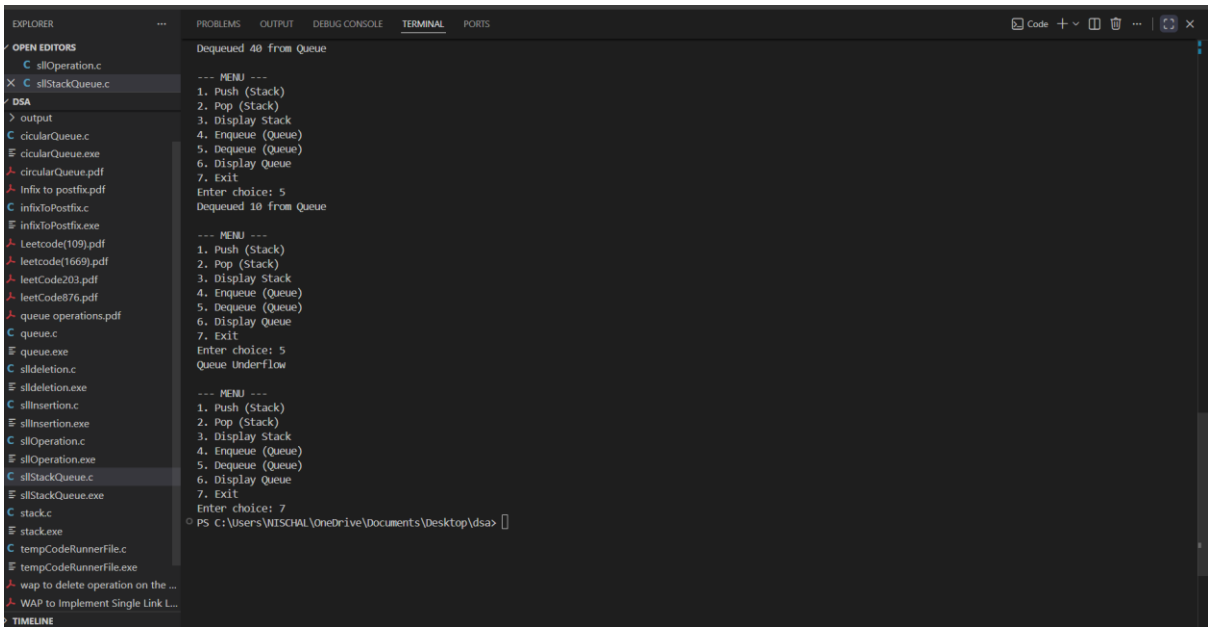
```
        case 2:
            pop();
            break;
        case 3:
            displayStack();
            break;
        case 4:
            printf("Enter value: ");
            scanf("%d", &value);
            enqueue(value);
            break;
        case 5:
            dequeue();
            break;
        case 6:
            displayQueue();
            break;
        case 7:
            exit(0);
        default:
            printf("Invalid Choice\n");
        }
    }
    return 0;
}
```

OUTPUT:


EXPLORER
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
OPEN EDITORS
sllOperation.c
sllStackQueue.c
DSA
output
cicularQueue.c
cicularQueue.exe
circularQueue.pdf
Infix to postfix.pdf
infixToPostfix.c
infixToPostfix.exe
Leetcode(109).pdf
leetcode(1669).pdf
leetCode203.pdf
leetCode876.pdf
queue operations.pdf
queue.c
queue.exe
slldeletion.c
slldeletion.exe
sllInsertion.c
sllInsertion.exe
sllOperation.c
sllOperation.exe
sllStackQueue.c
sllStackQueue.exe
stack.c
stack.exe
tempCodeRunnerFile.c
tempCodeRunnerFile.exe
wap to delete operation on the ...
WAP to Implement Single Link L...
TIMELINE
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa"
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa> cd "c:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa\" ; if ($?) { gcc sllStackQueue.c -o sllStackQueue } ; if ($?) { .
\sllStackQueue }
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 1
Enter value: 20
Pushed 20 into Stack
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 1
Enter value: 60
Pushed 60 into Stack
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 3
Stack elements:
60 20
--- MENU ---
1. Push (Stack)
dsa Debug Live Share Debug Compile Compile & Run
Ln 140, Col 1 (3149 selected) Spaces: 4 UTF-8 CRLF C Go Live windows-gcc-x86 Prettier


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Stack elements:
60 20
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 2
Popped 60 from Stack
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 4
Enter value: 40
Enqueued 40 into Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 4
Enter value: 10
Enqueued 10 into Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
dsa Debug Live Share Debug Compile Compile & Run
Ln 140, Col 1 (3149 selected) Spaces: 4 UTF-8 CRLF C Go Live windows-gcc-x86 Prettier


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 6
Queue elements:
40 10
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 5
Dequeued 40 from Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 5
Dequeued 10 from Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Live Share
Debug
Compile
Compile & Run
Ln 140, Col 1 (3149 selected)
Spaces: 4
UTF-8
CRLF
C
Go Live
windows-gcc-x86
Prettier


EXPLORER
OPEN EDITORS
sllOperation.c
sllStackQueue.c
DSA
output
cicularQueue.c
cicularQueue.exe
circularQueue.pdf
Infix to postfix.pdf
infixToPostfix.c
infixToPostfix.exe
Leetcode(109).pdf
leetcode(1669).pdf
leetCode203.pdf
leetCode876.pdf
queue operations.pdf
queue.c
queue.exe
sllDeletion.c
sllDeletion.exe
sllInsertion.c
sllInsertion.exe
sllOperation.c
sllOperation.exe
sllStackQueue.c
sllStackQueue.exe
stack.c
stack.exe
tempCodeRunnerFile.c
tempCodeRunnerFile.exe
wap to delete operation on the ...
WAP to Implement Single Link L...
TIMELINE
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
Dequeued 40 from Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 5
Dequeued 10 from Queue
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 5
Queue Underflow
--- MENU ---
1. Push (Stack)
2. Pop (Stack)
3. Display Stack
4. Enqueue (Queue)
5. Dequeue (Queue)
6. Display Queue
7. Exit
Enter choice: 7
PS C:\Users\NISCHAL\OneDrive\Documents\Desktop\dsa>