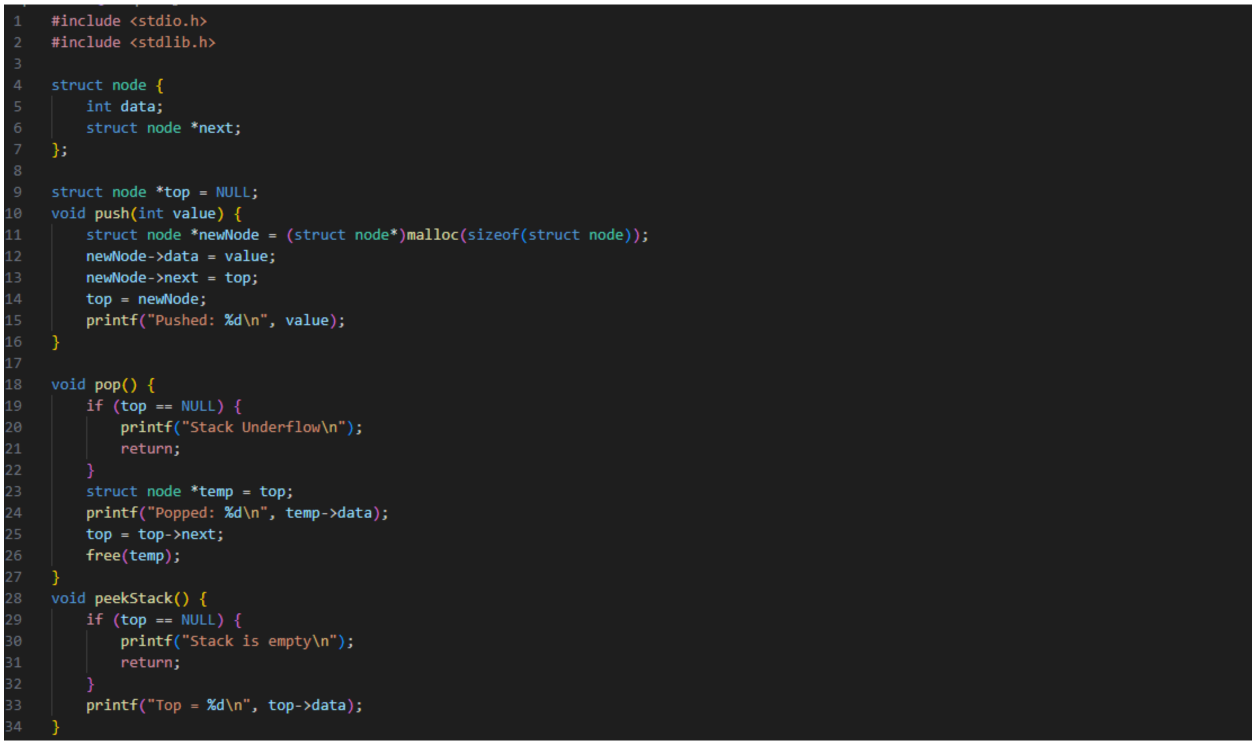
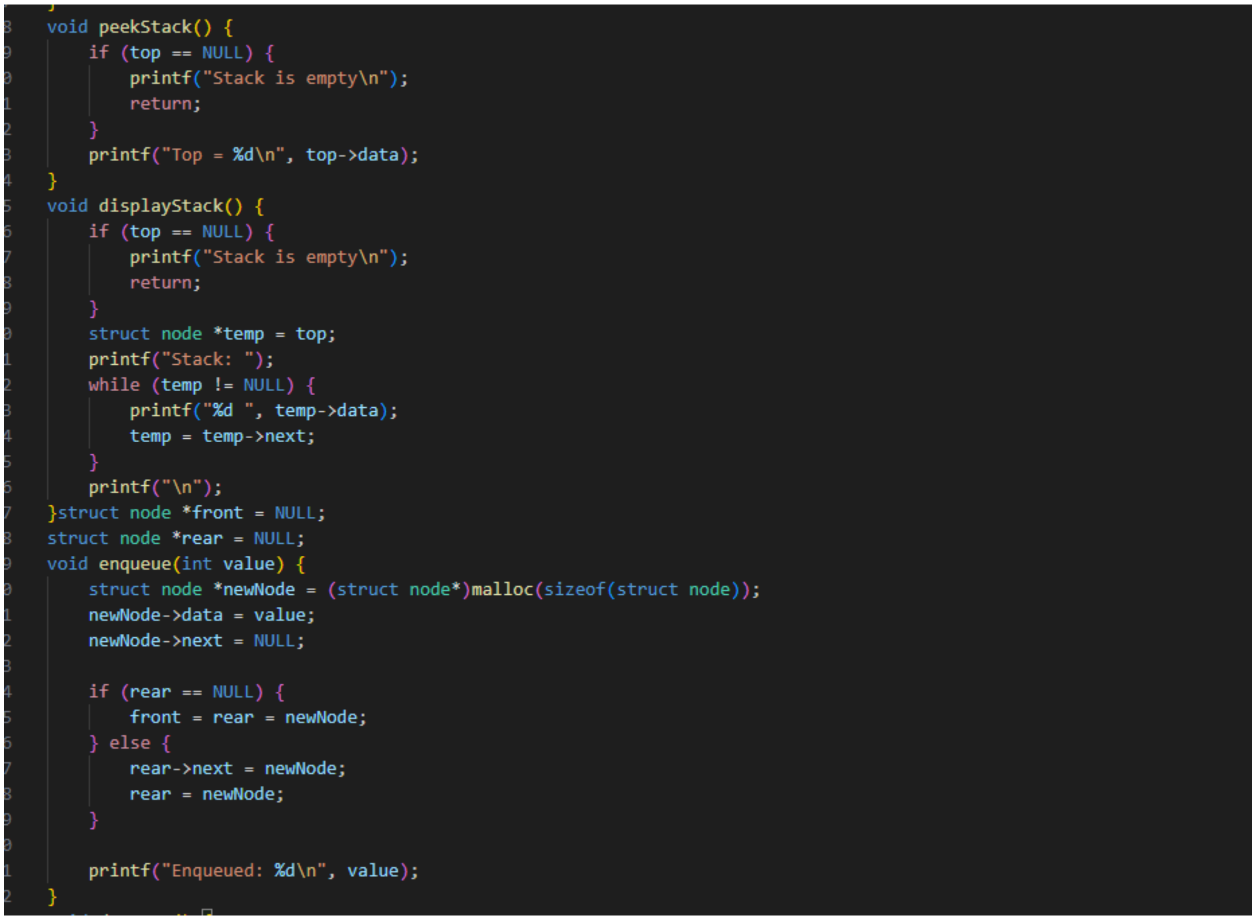
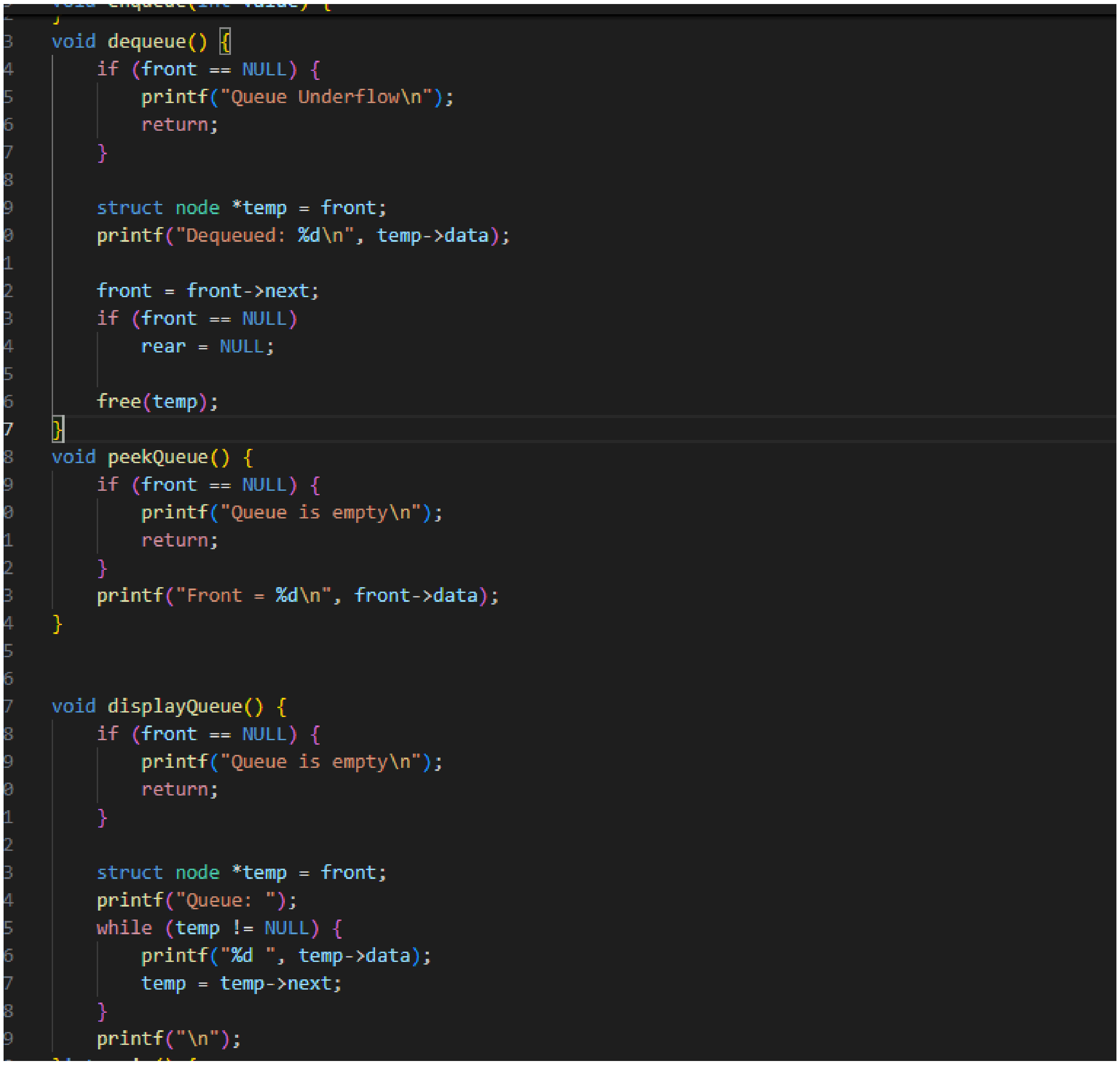


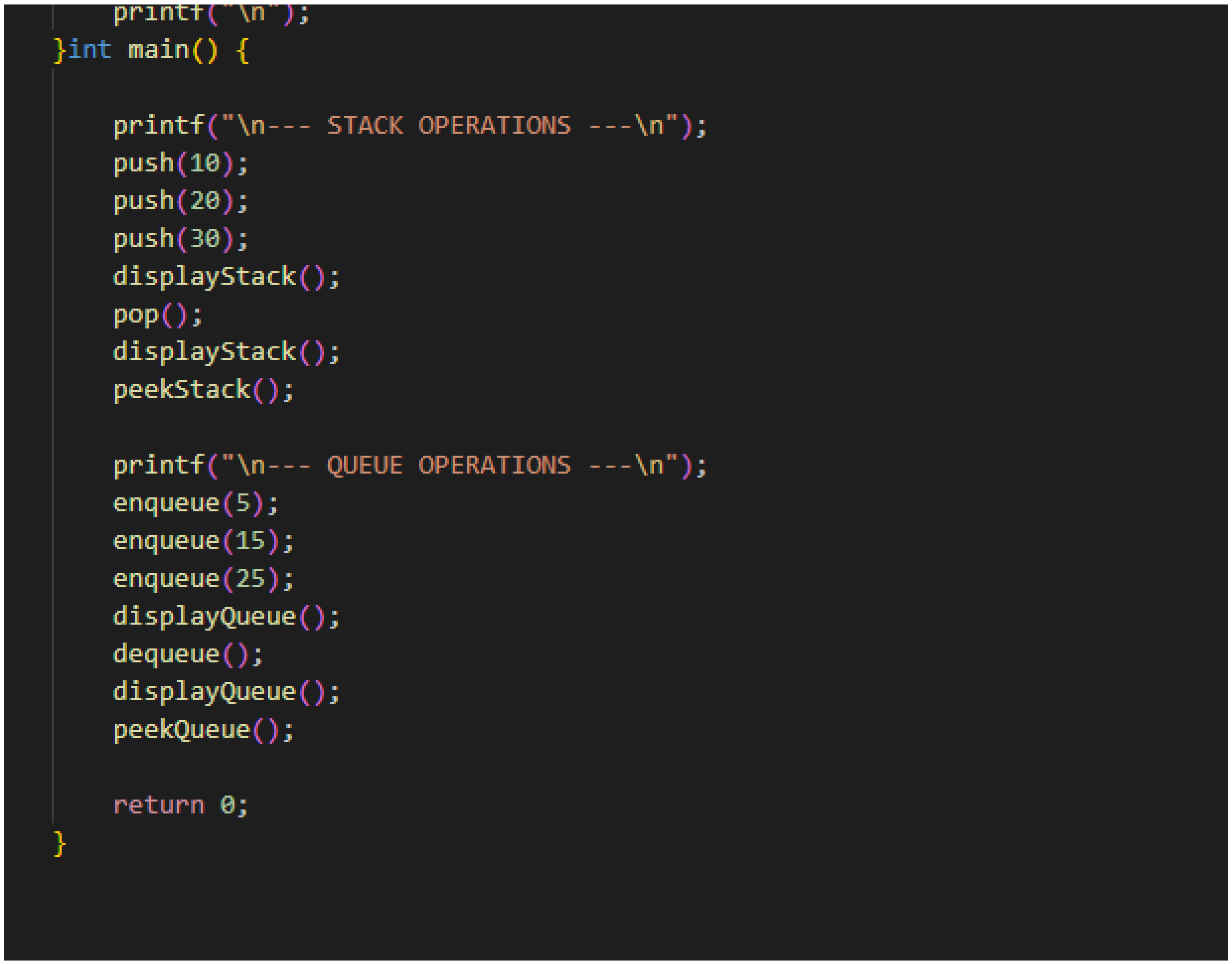
```c
#include <stdio.h>
#include <stdlib.h>

struct node {
    int data;
    struct node *next;
};

struct node *top = NULL;
void push(int value) {
    struct node *newNode = (struct node*)malloc(sizeof(struct node));
    newNode->data = value;
    newNode->next = top;
    top = newNode;
    printf("Pushed: %d\n", value);
}

void pop() {
    if (top == NULL) {
        printf("Stack Underflow\n");
        return;
    }
    struct node *temp = top;
    printf("Popped: %d\n", temp->data);
    top = top->next;
    free(temp);
}
void peekStack() {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    printf("Top = %d\n", top->data);
}
```



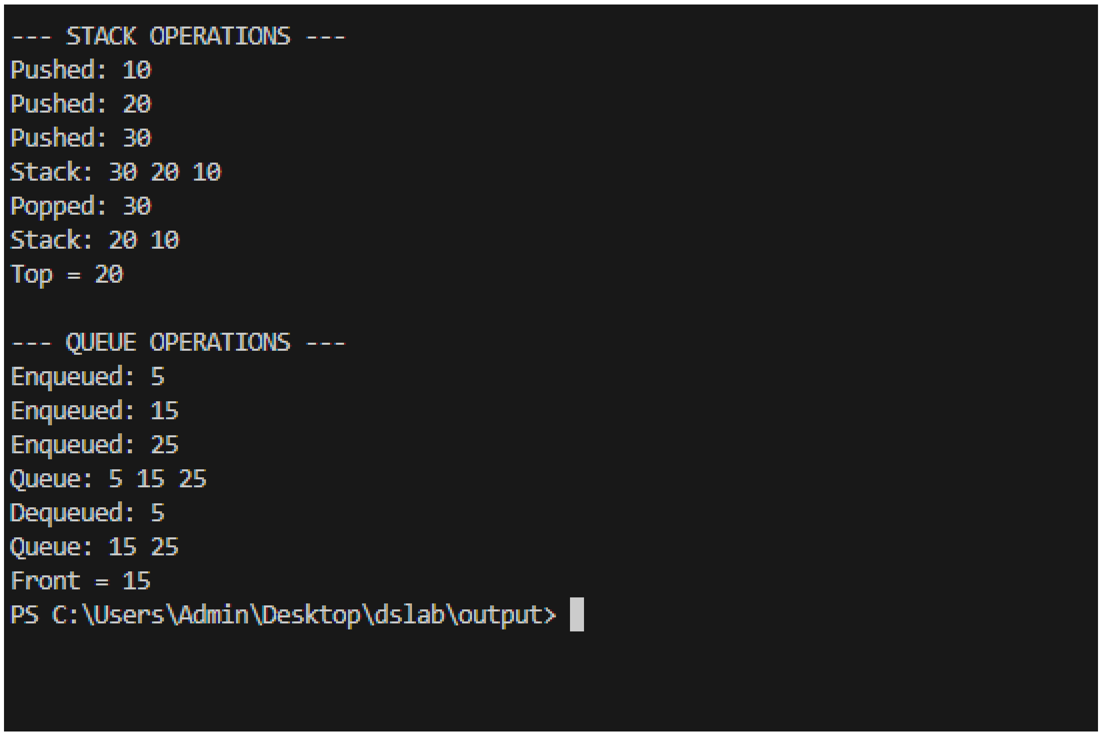
```c
void peekStack() {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    printf("Top = %d\n", top->data);
}
void displayStack() {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    struct node *temp = top;
    printf("Stack: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}struct node *front = NULL;
struct node *rear = NULL;
void enqueue(int value) {
    struct node *newNode = (struct node*)malloc(sizeof(struct node));
    newNode->data = value;
    newNode->next = NULL;

    if (rear == NULL) {
        front = rear = newNode;
    } else {
        rear->next = newNode;
        rear = newNode;
    }

    printf("Enqueued: %d\n", value);
}
```



```c
void dequeue() {
    if (front == NULL) {
        printf("Queue Underflow\n");
        return;
    }

    struct node *temp = front;
    printf("Dequeued: %d\n", temp->data);

    front = front->next;
    if (front == NULL)
        rear = NULL;

    free(temp);
}
void peekQueue() {
    if (front == NULL) {
        printf("Queue is empty\n");
        return;
    }
    printf("Front = %d\n", front->data);
}


void displayQueue() {
    if (front == NULL) {
        printf("Queue is empty\n");
        return;
    }

    struct node *temp = front;
    printf("Queue: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
```



```c
    printf("\n");
}int main() {

    printf("\n--- STACK OPERATIONS ---\n");
    push(10);
    push(20);
    push(30);
    displayStack();
    pop();
    displayStack();
    peekStack();

    printf("\n--- QUEUE OPERATIONS ---\n");
    enqueue(5);
    enqueue(15);
    enqueue(25);
    displayQueue();
    dequeue();
    displayQueue();
    peekQueue();

    return 0;
}
```



```
--- STACK OPERATIONS ---
Pushed: 10
Pushed: 20
Pushed: 30
Stack: 30 20 10
Popped: 30
Stack: 20 10
Top = 20

--- QUEUE OPERATIONS ---
Enqueued: 5
Enqueued: 15
Enqueued: 25
Queue: 5 15 25
Dequeued: 5
Queue: 15 25
Front = 15
PS C:\Users\Admin\Desktop\dslab\output>
```