

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct node {
5      int data;
6      struct node *next;
7  };
8
9  struct node *top = NULL;
10 void push(int value) {
11     struct node *newNode = (struct node*)malloc(sizeof(struct node));
12     newNode->data = value;
13     newNode->next = top;
14     top = newNode;
15     printf("Pushed: %d\n", value);
16 }
17
18 void pop() {
19     if (top == NULL) {
20         printf("Stack Underflow\n");
21         return;
22     }
23     struct node *temp = top;
24     printf("Popped: %d\n", temp->data);
25     top = top->next;
26     free(temp);
27 }
28 void peekStack() {
29     if (top == NULL) {
30         printf("Stack is empty\n");
31         return;
32     }
33     printf("Top = %d\n", top->data);
34 }

```

```

3 void peekStack() {
4     if (top == NULL) {
5         printf("Stack is empty\n");
6         return;
7     }
8     printf("Top = %d\n", top->data);
9 }
10 void displayStack() {
11     if (top == NULL) {
12         printf("Stack is empty\n");
13         return;
14     }
15     struct node *temp = top;
16     printf("Stack: ");
17     while (temp != NULL) {
18         printf("%d ", temp->data);
19         temp = temp->next;
20     }
21     printf("\n");
22 }
23 struct node *front = NULL;
24 struct node *rear = NULL;
25 void enqueue(int value) {
26     struct node *newNode = (struct node*)malloc(sizeof(struct node));
27     newNode->data = value;
28     newNode->next = NULL;
29
30     if (rear == NULL) {
31         front = rear = newNode;
32     } else {
33         rear->next = newNode;
34         rear = newNode;
35     }
36
37     printf("Enqueued: %d\n", value);
38 }

```

```

void enqueue(int value) {
}

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");
}

```

```

    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> █