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

// Structure for a node
struct Node {
    int data;
    struct Node* next;
};

// Function to create a new node
struct Node* createNode(int data) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

// Function to push an element into the stack
void push(struct Node** top, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = *top;
    *top = newNode;
    printf("Element %d pushed to stack\n", data);
}

// Function to pop an element from the stack
void pop(struct Node** top) {
    if (*top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    struct Node* temp = *top;
    *top = (*top)->next;
    printf("Element %d popped from stack\n", temp->data);
    free(temp);
}

// Function to display the stack
void displayStack(struct Node* top) {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    struct Node* temp = top;
    printf("Stack elements: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

// Function to enqueue an element into the queue
void enqueue(struct Node** front, struct Node** rear, int data) {
    struct Node* newNode = createNode(data);
    if (*front == NULL) {
        *front = *rear = newNode;
    } else {
        (*rear)->next = newNode;
        *rear = newNode;
    }
    printf("Element %d enqueued to queue\n", data);
}

// Function to dequeue an element from the queue
void dequeue(struct Node** front, struct Node** rear) {
    if (*front == NULL) {
        printf("Queue is empty\n");
        return;
    }
    struct Node* temp = *front;
    *front = (*front)->next;
    if (*front == NULL) {
        *rear = NULL;
    }
    printf("Element %d dequeued from queue\n", temp->data);
    free(temp);
}

// Function to display the queue
void displayQueue(struct Node* front) {
    if (front == NULL) {
        printf("Queue is empty\n");
        return;
    }
    struct Node* temp = front;
    printf("Queue elements: ");
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}
```

```
int main() {
    struct Node* stackTop = NULL;
    struct Node* queueFront = NULL;o
    struct Node* queueRear = NULL;

    int choice, data;

    do {
        printf("1. Push element onto stack\n");
        printf("2. Pop element from stack\n");
        printf("3. Display stack\n");
        printf("4. Enqueue element into queue\n");
        printf("5. Dequeue element from queue\n");
        printf("6. Display queue\n");
        printf("0. Exit\n");
        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice) {
            case 1:
                printf("Enter element to push onto stack: ");
                scanf("%d", &data);
                push(&stackTop, data);
                break;
            case 2:
                pop(&stackTop);
                break;
            case 3:
                displayStack(stackTop);
                break;
            case 4:
                printf("Enter element to enqueue into queue: ");
                scanf("%d", &data);
                enqueue(&queueFront, &queueRear, data);
                break;
            case 5:
                dequeue(&queueFront, &queueRear);
                break;
            case 6:
                displayQueue(queueFront);
                break;
            case 0:
                printf("Exiting...\n");
                break;
            default:
                printf("Invalid choice\n");
        }
        printf("\n");
    } while (choice != 0);

    return 0;
}
```