Question:

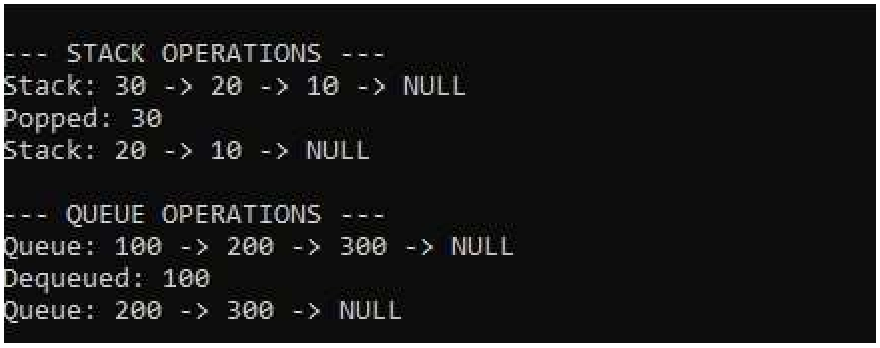
WAP to Implement Single Link List to simulate Stack &
Queue Operations.

Input:

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

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

struct Node* createNode(int data) {
    struct Node *newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = data;
    newNode->next = NULL;
    return newNode;
}

void push(struct Node **top, int value) {
    struct Node *newNode = createNode(value);
    newNode->next = *top;
    *top = newNode;
}

int pop(struct Node **top) {
    if (*top == NULL) {
        printf("Stack Underflow!\n");
        return -1;
    }
    struct Node *temp = *top;
    int val = temp->data;
    *top = (*top)->next;
    free(temp);
    return val;
}

void displayStack(struct Node *top) {
    printf("Stack: ");
    while (top != NULL) {
```

```c
        printf("%d -> ", top->data);
        top = top->next;
    }
    printf("NULL\n");
}


void enqueue(struct Node **front, struct Node **rear, int value) {
    struct Node *newNode = createNode(value);
    if (*rear == NULL) {
        *front = *rear = newNode;
        return;
    }
    (*rear)->next = newNode;
    *rear = newNode;
}

int dequeue(struct Node **front, struct Node **rear) {
    if (*front == NULL) {
        printf("Queue Underflow!\n");
        return -1;
    }

    struct Node *temp = *front;
    int val = temp->data;
    *front = (*front)->next;

    if (*front == NULL)
        *rear = NULL;

    free(temp);
    return val;
}

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

```c
int main() {

    struct Node *stack = NULL;
    struct Node *front = NULL;
    struct Node *rear = NULL;

    printf("\n--- STACK OPERATIONS ---\n");
    push(&stack, 10);
    push(&stack, 20);
    push(&stack, 30);
    displayStack(stack);

    printf("Popped: %d\n", pop(&stack));
    displayStack(stack);

    printf("\n--- QUEUE OPERATIONS ---\n");
    enqueue(&front, &rear, 100);
    enqueue(&front, &rear, 200);
    enqueue(&front, &rear, 300);
    displayQueue(front);

    printf("Dequeued: %d\n", dequeue(&front, &rear));
    displayQueue(front);

    return 0;
}
```



```
--- STACK OPERATIONS ---
Stack: 30 -> 20 -> 10 -> NULL
Popped: 30
Stack: 20 -> 10 -> NULL

--- QUEUE OPERATIONS ---
Queue: 100 -> 200 -> 300 -> NULL
Dequeued: 100
Queue: 200 -> 300 -> NULL
```