Lab 6b) WAP to Implement Single Link List to simulate Stack & Queue Operations.

code:

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

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

Struct node * push (structnode * top, int x) {
    Struct node * p = malloc (sizeof (struct node));
    p -> data = x;
    p -> next = top;
    return p;
}
Struct node * pop (struct node * top) {
    if (top) {
        printf ("popped %d\n", top -> data);
        Struct node * t = top;
        top = top -> next;
        free (t);
    } else {
        printf ("stack is empty\n");
    }
    return top;

struct node * enqueue (struct node * rear, stryct
                                node** front, int
    struct node * p = malloc (size of (struct node));
    p -> data = x;
    p -> next = NULL;
```

```
if (front ...                    ...);
else rear -> next = p;
printf ("Enqueued %d \n", x);
return p;

struct node* dequeue (struct node * front) {
    if (front) {
        printf ("Dequeued %d \n", front->data);
        struct node *t =front;
        front = front -> next;
        free (t);
    } else {
        printf (" Queue is empty \n ");
    }
    return front;

void display (struct node * head) {
    if (head == NULL ) {
        printf ("Empty \n");
        return ;
    }
    while (head ) {
        printf ("%d", head -> data);
        head = head -> next;
    }
    print f ("\n ");
}
```

```
int main (){
    struct node * top = NULL, * front = NULL, * rear
                                              =Nu
    int c, x;
printf ("\n MENU \n");
printf ("1. Stack - Push \n");
printf ("2. Stack - pop \n");
printf ("3. Queue - Enqueue \n");
printf ("4. Queue - Dequeue \n");
printf ("5. Stack - Display \n");
printf ("6. Queue - Display \n");
printf ("7- Exit \n");
printf ("\n ");
while (1){
    printf ("\n Enter choice : ");
    scanf ("%d ", &c);
    switch (c) {
        case 1:
            printf ("Enter value to push: ");
            scanf ("%d ", &x);
            top = push (top, x);
            printf ("Pushed %d \n", x);
            break;
        case 2:
            top = pop (top);
            break;
        case 3:
            printf (" Enter value to enqueue : ");
            scanf ("%d", &x);
```

```
break;
case 4:
    front = dequeue (front);
    if (front == NULL) rear = NULL;
    break;
case 5:
    printf ("Stack contents: ");
    display (top);
    break;
    case 6:
        printf ("Queue contents: ");
        display (front);
        break;
    case 7:
        printf ("Exiting program.\n");
        return 0;
    default:
        printf ("Invalid choice. Try again.\n");
    }
  }
}
```

Output:

MENU
1. Stack - Push
2. Stack - POP
3. Queue - Enqueue
4. Queue - Dequeue
5. Stack - Display
6. Queue - Display
7. Exit

MG
1/12/25.

Enter choice : 5

stack contents : Empty

Enter choice : 6

Queue contents : Empty

Enter choice : 1

Enter value to push : 1

Pushed 1.

Enter choice : 1

Enter value to push : 2

Pushed 2.

Enter choice : 1

Enter value to push : 3

Pushed 3

Enter choice : 2

popped 3

Enter choice : 5

Stack contents : 2 1

Enter choice : 3

Enter value to enqueue : 4

Enqueued 4

Enter choice : 3

Enter value to enqueue : 5

Enqueued 5

Enter value to enqueue : 6

Enqueued 6.

Enter choice : 4

Dequeued 4

Enter choice : 6

Queue contents : 5 6

Enter choice : 7

Exiting program.