EXPERIMENT NUMBER 3

Aim:

Create a stack and perform:

a). Push

b). Pop

c). Peek

d). Traverse

using linked list.

Theory:

A stack is a data structure that stores elements in a last-in, first-out (LIFO) order.

Push: Returns the new top of the stack after adding an element.

Pop: Returns the new top of the stack after removing an element.

Peek: Returns the top element without modifying the stack.

Traverse: Displays all the elements in the stack.

Code:

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;
}
```

```
struct Node* push(struct Node* top, int data) {
    struct Node* newNode = createNode(data);
    newNode->next = top;
    top = newNode;
    printf("%d pushed to stack\n", data);
    return top;
}

struct Node* pop(struct Node* top) {
    if (top == NULL) {
        printf("Stack Underflow\n");
        return NULL;
    }
    int poppedData = top->data;
    struct Node* temp = top;
    top = top->next;
    free(temp);
    printf("%d popped from stack\n", poppedData);
    return top;
}

int peek(struct Node* top) {
    if (top == NULL) {
        printf("Stack is empty\n");
        return -1;
    }
    return top->data;
}

void traverse(struct Node* top) {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    struct Node* temp = top;
    while (temp != NULL) {
        printf("%d -> ", temp->data);
        temp = temp->next;
    }
    printf("NULL\n");
}

int main() {
    struct Node* stack = NULL;

    stack = push(stack, 10);
    stack = push(stack, 20);
    stack = push(stack, 30);
    traverse(stack);

    printf("Top element is %d\n", peek(stack));

    stack = pop(stack);
    traverse(stack);

    stack = pop(stack);
    traverse(stack);

    return 0;
}
```

Output:

```
10 pushed to stack

20 pushed to stack

30 pushed to stack

30 -> 20 -> 10 -> NULL

Top element is 30

30 popped from stack

20 -> 10 -> NULL

20 popped from stack

10 -> NULL
```