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

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

struct Node* top = NULL;

void linkedListTraversal(struct Node *ptr)
{
    while (ptr != NULL)
    {
        printf("Element: %d\n", ptr->data);
        ptr = ptr->next;
    }
}

int isEmpty(struct Node* top){
    if (top==NULL){
        return 1;
    }
    else{
        return 0;
    }
}

int isFull(struct Node* top){
    struct Node* p = (struct Node*)malloc(sizeof(struct Node));
    if(p==NULL){
        return 1;
    }
    else{
        return 0;
    }
}

struct Node* push(struct Node* top, int x){
```

```
40:     if(isFull(top)){
41:         printf("Stack Overflow\n");
42:     }
43:     else{
44:         struct Node* n = (struct Node*) malloc(sizeof(struct Node))
45:         n->data = x;
46:         n->next = top;
47:         top = n;
48:         return top;
49:     }
50: }
51:
52: int pop(struct Node*tp){
53:     if(isEmpty(tp)){
54:         printf("Stack Underflow\n");
55:     }
56:     else{
57:         struct Node* n = tp;
58:         top = (tp)->next;
59:         int x = n->data;
60:         free(n);
61:         return x;
62:     }
63: }
64:
65: int main(){
66:     top = push(top, 78);
67:     top = push(top, 7);
68:     top = push(top, 8);
69:     linkedListTraversal(top);
70:
71:     int element = pop(top);
72:     printf("Popped element is %d\n", element);
73:     linkedListTraversal(top);
74:
75:
76:
77:     return 0;
78: }
```

79: