Program for stack using array:

```
#include <stdio.h>
#define MAX_SIZE 100
int stack[MAX_SIZE];
int top = -1;
void push(int value) {
    if (top == MAX_SIZE - 1) {
        printf("Stack Overflow\n");
        return;
    }
    stack[++top] = value;
}
void pop() {
    if (top == -1) {
        printf("Stack Underflow\n");
        return;
    }
    top--;
}
int peek() {
    if (top == -1) {
        printf("Stack is empty\n");
        return -1;
    }
    return stack[top];
}
int main() {
    push(1);
    push(2);
    push(3);
    printf("Top element: %d\n", peek());
    pop();
    printf("Top element after pop: %d\n", peek());
    return 0;
}
```

Output:

```
Top element: 30
Top element after pop: 20
```

Program for stack using linked list:

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

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

struct Node* top = NULL;

void push(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = top;
    top = newNode;
}

void pop() {
    if (top == NULL) {
        printf("Stack is empty\n");
        return;
    }
    struct Node* temp = top;
    top = top->next;
    free(temp);
}

void display() {
    struct Node* temp = top;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}

int main() {
    push(1);
    push(2);
    push(3);
    display();
    pop();
    display();
    return 0;}
```

Output:

```
Stack elements: 3 2 1
Stack elements: 2 1
```