- **Stack using array:**

```
#include <stdio.h>
#define MAX_SIZE 10
typedef struct {
    int data[MAX_SIZE];
    int top;
} Stack;
Stack* createStack() {
    Stack* stack = (Stack*) malloc(sizeof(Stack));
    stack->top = -1;
    return stack;
}
int isEmpty(Stack* stack) {
    return stack->top == -1;
}
void push(Stack* stack, int value) {
    if (stack->top < MAX_SIZE - 1) {
        stack->data[++stack->top] = value;
    } else {
        printf("Stack overflow!\n");
    }
}
int pop(Stack* stack) {
    if (!isEmpty(stack)) {
        return stack->data[stack->top--];
    } else {
        printf("Stack underflow!\n");
        return -1;
    }
}
void printStack(Stack* stack) {
for (int i = 0; i <= stack->top; i++) {
```

```
        printf("%d ", stack->data[i]);
    }
    printf("\n");
}
int main() {
    Stack* stack = createStack();
    push(stack, 10);
    push(stack, 20);
    push(stack, 30);
    printStack(stack); // Output: 10 20 30
    printf("Popped element: %d\n", pop(stack)); // Output: 30
    printStack(stack); // Output: 10 20
    return 0;
}
```

- **Out put:**
  Stack elements: 10,20,30
  Popped element: 30
  Stack elements after pop: 10,20
- **Stack using linked list:**

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
} Node;
typedef struct {
    Node* top;
} Stack;
Node* createNode(int value) {
    Node* node = (Node*) malloc(sizeof(Node));
    node->data = value;
    node->next = NULL;
    return node;
}
Stack* createStack() {
    Stack* stack = (Stack*) malloc(sizeof(Stack));
    stack->top = NULL;
    return stack;
}
int isEmpty(Stack* stack) {
    return stack->top == NULL;
}
void push(Stack* stack, int value) {
    Node* node = createNode(value);
    node->next = stack->top;
```

```
    stack->top = node;
}
int pop(Stack* stack) {
    if (!isEmpty(stack)) {
        int value = stack->top->data;
        Node* temp = stack->top;
        stack->top = stack->top->next;
        free(temp);
        return value;
    } else {
        printf("Stack underflow!\n");
        return -1;
    }
}
void printStack(Stack* stack) {
    Node* temp = stack->top;
    while (temp != NULL) {
        printf("%d ", temp->data);
        temp = temp->next;
    }
    printf("\n");
}
int main() {
    Stack* stack = createStack();
    push(stack, 10);
    push(stack, 20);
    push(stack, 30);
    printf("Stack elements: ");
    printStack(stack); // Output: 30 20 10
    printf("Popped element: %d\n", pop(stack)); // Output: 30
    printf("Stack elements after pop: ");
    printStack(stack); // Output: 20 10
    return 0;
}
```

- **Output:**

Stack elements: 30,20,10
Popped element: 30
Stack elements after pop: 20,10