1) Write a program for stack using array n linked lis t(operations).

Code

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
} Node;
typedef struct {
    Node* top;
} Stack;

void init(Stack *s) {
    s->top = NULL;
}
int isEmpty(Stack *s) {
    return s->top == NULL;
}
void push(Stack *s, int value) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    if (!newNode) {
        printf("Memory allocation error\n");
        return;
    }
    newNode->data = value;
    newNode->next = s->top;
    s->top = newNode;
}
int pop(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack underflow\n");
        return -1;
    }
    Node* temp = s->top;
    int value = temp->data;
    s->top = temp->next;
    free(temp);
    return value;
}
```

```
int peek(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack is empty\n");
        return -1;
    }
    return s->top->data;
}
void printStack(Stack *s) {
    Node* current = s->top;
    if (isEmpty(s)) {
        printf("Stack is empty\n");
        return;
    }
    while (current) {
        printf("%d\n", current->data);
        current = current->next;
    }
}
int main() {
    Stack s;
    init(&s);
    push(&s, 15);
    push(&s, 20);
    push(&s, 25);
    printf("peek element: %d\n", peek(&s));
    printStack(&s);
    printf("Pop element: %d\n", pop(&s));
    printStack(&s);
    return 0;
}
```

Output:

```
peek element: 25
 25
 20
 15
Pop element: 25
 20
 15
```

2)write a program for stack using n linked list(operations).

Code

```
#include <stdio.h>
#include <stdlib.h>
typedef struct Node {
    int data;
    struct Node* next;
} Node;
typedef struct {
    Node* top;
} Stack;
void init(Stack *s) {
    s->top = NULL;
}
int isEmpty(Stack *s) {
    return s->top == NULL;
}
void push(Stack *s, int value) {
    Node* newNode = (Node*)malloc(sizeof(Node));
    if (newNode == NULL) {
        printf("Memory allocation error\n");
        return;
    }
    newNode->data = value;
    newNode->next = s->top;
    s->top = newNode;
}
int pop(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack underflow\n");
        return -1; // Return a special value to indicate underflow
    }
    Node* temp = s->top;
    int value = temp->data;
    s->top = temp->next;
    free(temp);
    return value;
}
int peek(Stack *s) {
    if (isEmpty(s)) {
        printf("Stack is empty\n");
        return -1; // Return a special value to indicate empty stack
    }
    return s->top->data;
}

void printStack(Stack *s) {
    Node* current = s->top;
    if (isEmpty(s)) {
        printf("Stack is empty\n");
        return;
    }
    printf("Stack elements:\n");
    while (current) {
        printf("%d\n", current->data);
        current = current->next;
    }
}
int main() {
    Stack s;
    init(&s);
    push(&s, 50);
    push(&s, 60);
    push(&s, 70);
    printf("peek element: %d\n", peek(&s));
    printStack(&s);
    printf("Pop element: %d\n", pop(&s));
    printStack(&s);
    while (!isEmpty(&s)) {
        pop(&s);
    }
    return 0;
}
```

Output:

```
peek element: 70
Stack elements:
70
60
50
Pop element: 70
Stack elements:
60
50
```