Stack Using Array

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 100
int stack[MAX];
int top = -1;
void push(int value) {
   if (top >= MAX - 1) {
      printf("Stack Overflow\n");
   } else {
      stack[++top] = value;
      printf("%d pushed into stack\n", value);
   }
}
int pop() {
   if (top < 0) {
      printf("Stack Underflow\n");
      return -1;
   } else {
      int value = stack[top--];
      return value;
   }
}
int peek() {
   if (top < 0) {
      printf("Stack is empty\n");
      return -1;
   } else {
      return stack[top];
   }
```

```
}
int isEmpty() {
    return top == -1;
}
void display() {
    if (top < 0) {
        printf("Stack is empty\n");
    } else {
        printf("Stack elements are:\n");
        for (int i = top; i >= 0; i--) {
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
}
int main() {
    push(10);
    push(20);
    push(30);
    display();
    printf("Top element is %d\n", peek());
    printf("%d popped from stack\n", pop());
    display();
    printf("Stack is empty: %s\n", isEmpty() ? "Yes" : "No");
    return 0;
}
```

Output:

```
10 pushed onto stack
20 pushed onto stack
30 pushed onto stack
```

Stack elements are:

30 20 10

Top element is 30

30 popped from the stack

Stack elements are:

20 10

Stack is empty: No

Stack Using Linked List

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
    int data;
    struct Node* next;
};
struct Node* createNode(int value) {
    struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
    newNode->data = value;
    newNode->next = NULL;
    return newNode;
}
void push(struct Node** top, int value) {
    struct Node* newNode = createNode(value);
    newNode->next = *top;
    *top = newNode;
    printf("%d pushed onto stack\n", value);
}
int pop(struct Node** top) {
    if (*top == NULL) {
        printf("Stack Underflow\n");
```

```
        return -1;
    } else {
        struct Node* temp = *top;
        int poppedValue = temp->data;
        *top = (*top)->next;
        free(temp);
        return poppedValue;
    }
}
int peek(struct Node* top) {
    if (top == NULL) {
        printf("Stack is empty\n");
        return -1;
    } else {
        return top->data;
    }
}
int isEmpty(struct Node* top) {
    return top == NULL;
}
void display(struct Node* top) {
    if (top == NULL) {
        printf("Stack is empty\n");
    } else {
        struct Node* temp = top;
        printf("Stack elements are:\n");
        while (temp != NULL) {
            printf("%d ", temp->data);
            temp = temp->next;
        }
        printf("\n");
```

```
    }
}
int main() {
    struct Node* stack = NULL;
    push(&stack, 10);
    push(&stack, 20);
    push(&stack, 30);
    display(stack);
    printf("Top element is %d\n", peek(stack));
    printf("%d popped from stack\n", pop(&stack));
    display(stack);

    printf("Stack is empty: %s\n", isEmpty(stack) ? "Yes" : "No");
    return 0;
}
```

Output:

```
10 pushed onto stack
20 pushed onto stack
30 pushed onto stack
Stack elements are:
30 20 10
Top element is 30
30 popped from the stack
Stack elements are:
20 10
Stack is empty: No
```