SEM 3\Exp5\Stack_ARR.c

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

#define MAX 100 // Maximum size of the stack

// Stack structure using arrays
struct StackArray
{
    int top;
    int arr[MAX];
};

// Function to create a stack
struct StackArray *createStack()
{
    struct StackArray *stack = (struct StackArray *)malloc(sizeof(struct StackArray));
    stack->top = -1; // Initialize the top index
    return stack;
}

// Check if the stack is full
int isFull(struct StackArray *stack)
{
    return stack->top == MAX - 1;
}

// Check if the stack is empty
int isEmpty(struct StackArray *stack)
{
    return stack->top == -1;
}

// Push an element onto the stack
void push(struct StackArray *stack, int value)
{
    if (isFull(stack))
    {
        printf("Stack overflow!\n");
        return;
    }
    stack->arr[++stack->top] = value;
    printf("%d pushed onto stack\n", value);
}

// Pop an element from the stack
int pop(struct StackArray *stack)
{
    if (isEmpty(stack))
    {
        printf("Stack underflow!\n");
        return -1;
```

```c
    }
    return stack->arr[stack->top--];
}

// Peek at the top element of the stack
int peek(struct StackArray *stack)
{
    if (isEmpty(stack))
    {
        printf("Stack is empty!\n");
        return -1;
    }
    return stack->arr[stack->top];
}

// Display the stack
void display(struct StackArray *stack)
{
    if (isEmpty(stack))
    {
        printf("Stack is empty!\n");
        return;
    }
    printf("Stack elements: ");
    for (int i = stack->top; i >= 0; i--)
    {
        printf("%d ", stack->arr[i]);
    }
    printf("\n");
}

int main()
{
    struct StackArray *stack = createStack();
    int choice, value;

    printf("\nStack Operations (Array Implementation):\n");
    printf("1. Push\n");
    printf("2. Pop\n");
    printf("3. Peek\n");
    printf("4. Display\n");
    printf("5. Exit\n");

    while (1)
    {

        printf("Enter your choice: ");
        scanf("%d", &choice);

        switch (choice)
        {
        case 1:
            printf("Enter the value to push: ");
            scanf("%d", &value);
```

```c
            push(stack, value);
            printf("\n");
            break;
        case 2:
            value = pop(stack);
            if (value != -1)
                printf("Popped value: %d\n", value);

            printf("\n");
            break;
        case 3:
            value = peek(stack);
            if (value != -1)
                printf("Top value: %d\n", value);

            printf("\n");
            break;
        case 4:
            display(stack);
            printf("\n");
            break;
        case 5:
            free(stack);
            exit(0);
        default:
            printf("Invalid choice!\n");
        }
    }

    return 0;
}

```