1. Write a program to simulate the working of stack using an array with the following:

a) Push

b) Pop

c) Display

The program should print appropriate messages for stack overflow, stack underflow

```
#include <stdio.h>
#define MAX 5  // Maximum size of stack

// Declare stack array and top pointer
int stack[MAX];
int top = -1;

// Function to check if the stack is full
int isFull() {
    if(top == MAX - 1)
        return 1;
    return 0;
}

// Function to check if the stack is empty
int isEmpty() {
    if(top == -1)
        return 1;
    return 0;
}

// Function to push an element onto the stack
void push(int value) {
    if(isFull()) {
        printf("Stack Overflow! Cannot push %d\n", value);
    } else {
```

```
        top++;
        stack[top] = value;
        printf("Pushed %d onto stack\n", value);
    }
}

// Function to pop an element from the stack
int pop() {
    if(isEmpty()) {
        printf("Stack Underflow! No elements to pop\n");
        return -1;
    } else {
        int value = stack[top];
        top--;
        return value;
    }
}

// Function to display the elements of the stack
void display() {
    if(isEmpty()) {
        printf("Stack is empty!\n");
    } else {
        printf("Stack elements: ");
        for(int i = top; i >= 0; i--) {
            printf("%d ", stack[i]);
        }
        printf("\n");
    }
}

int main() {
```

```
int choice, value;

while(1) {
    // Menu
    printf("\nStack Operations Menu:\n");
    printf("1. Push\n");
    printf("2. Pop\n");
    printf("3. Display\n");
    printf("4. Exit\n");
    printf("Enter your choice: ");
    scanf("%d", &choice);

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

        case 2:
            value = pop();
            if(value != -1) {
                printf("Popped %d from stack\n", value);
            }
            break;

        case 3:
            display();
            break;

        case 4:
            printf("Exiting...\n");
```

```
            return 0;

        default:
            printf("Invalid choice! Please try again.\n");
      }
   }

   return 0;
}
```

```
Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 10
Pushed 10 onto stack

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 20
Pushed 20 onto stack

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 30
Pushed 30 onto stack

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Popped 30 from stack

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 30
Invalid choice! Please try again.

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 3
Stack elements: 20 10

Stack Operations Menu:
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 4
Exiting...

Process returned 0 (0x0)   execution time : 29.019 s
Press any key to continue.
```