To implement basic stack operations using an array in C.

OK, here's the C code for implementing stack operations, along with the procedure and aim:

Aim

To implement basic stack operations using an array in C.

Procedure

1. **Define the stack:**
    - Declare an array to store the stack elements.
    - Declare a variable `top` to keep track of the top element's index. Initialize `top` to -1, indicating an empty stack.
2. **Implement `push`:**
    - Check if the stack is full (i.e., `top` is at the maximum array index).
    - If not full:
        - Increment `top`.
        - Store the new element at `stack[top]`.
3. **Implement `pop`:**
    - Check if the stack is empty (i.e., `top` is -1).
    - If not empty:
        - Retrieve the element at `stack[top]`.
        - Decrement `top`.
        - Return the retrieved element.
4. **Implement `display` (optional):**
    - Check if the stack is empty.
    - If not empty, print the elements from top to bottom.
5. **Main function:**
    - Demonstrate the stack operations by calling `push` and `pop`.
    - Optionally, call `display` to visualize the stack contents.

Code

C

```
#include <stdio.h>

#define MAX_SIZE 100 // Adjust as needed
```

```
int stack[MAX_SIZE];
int top = -1;

// Push an element onto the stack
void push(int item) {
    if (top >= MAX_SIZE - 1) {
        printf("Error: Stack Overflow\n");
    } else {
        stack[++top] = item;
        printf("%d pushed to the stack\n", item);
    }
}

// Pop an element from the stack
int pop() {
    if (top < 0) {
        printf("Error: Stack Underflow\n");
        return -1; // Or handle the error appropriately in your application
    } else {
        int item = stack[top--];
        printf("%d popped from the stack\n", item);
        return item;
    }
}

// Display the stack contents (optional)
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\n", stack[i]);
        }
    }
}

int main() {
    push(10);
    push(20);
    push(30);

    printf("Popped element: %d\n", pop());
```

```
    printf("Popped element: %d\n", pop());

    display(); // Display the remaining stack contents

    return 0;
}
```

Let me know if you'd like to explore more advanced stack operations or implementations!