**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

int stack[MAX];
int top = -1;

void push(int value) {
   if (top == MAX - 1) {
      printf("Stack Overflow! Cannot push %d\n", value);
   } else {
      top++;
      stack[top] = value;
      printf("%d pushed onto the stack.\n", value);
   }
}

void pop() {
   if (top == -1) {
      printf("Stack Underflow! Stack is empty.\n");
```

```
    } else {
        printf("Popped element: %d\n", stack[top]);
        top--;
    }
}

void display() {
    if (top == -1) {
        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) {
        printf("\n--- Stack Operations ---\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:
                pop();
                break;
            case 3:
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice! Please try again.\n");
        }
    }
    return 0;
}
```

**Output**

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

--- Stack Operations ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 20
20 pushed onto the stack.
```

```
--- Stack Operations ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 1
Enter value to push: 30
30 pushed onto the stack.

--- Stack Operations ---
1. Push
2. Pop
3. Display
4. Exit
Enter your choice: 2
Popped element: 30
```

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

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