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

#define MAX 100

// Stack structure
struct Stack {
   int items[MAX];
   int top;
};

// Function to initialize stack
void initStack(struct Stack* stack) {
   stack->top = -1;
}

// Function to check if stack is full
int isFull(struct Stack* stack) {
   return stack->top == MAX - 1;
}

// Function to check if stack is empty
int isEmpty(struct Stack* stack) {
   return stack->top == -1;
}

// Function to push item to stack
void push(struct Stack* stack, int value) {
   if (isFull(stack)) {
      printf("Stack overflow\n");
      return;
```

```
    }
    stack->items[++stack->top] = value;
}

// Function to pop item from stack
int pop(struct Stack* stack) {
    if (isEmpty(stack)) {
        printf("Stack underflow\n");
        exit(1);
    }
    return stack->items[stack->top--];
}

// Function to reverse an array using stack
void reverseArray(int arr[], int size) {
    struct Stack stack;
    initStack(&stack);

    // Push all elements to the stack
    for (int i = 0; i < size; i++) {
        push(&stack, arr[i]);
    }

    // Pop all elements back into the array (in reverse)
    for (int i = 0; i < size; i++) {
        arr[i] = pop(&stack);
    }
}

// Main function
int main() {
    int arr[] = {1, 2, 3, 4, 5};
    int size = sizeof(arr) / sizeof(arr[0]);

    printf("Original Array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    reverseArray(arr, size);

    printf("\nReversed Array: ");
    for (int i = 0; i < size; i++) {
        printf("%d ", arr[i]);
    }

    return 0;
}
```