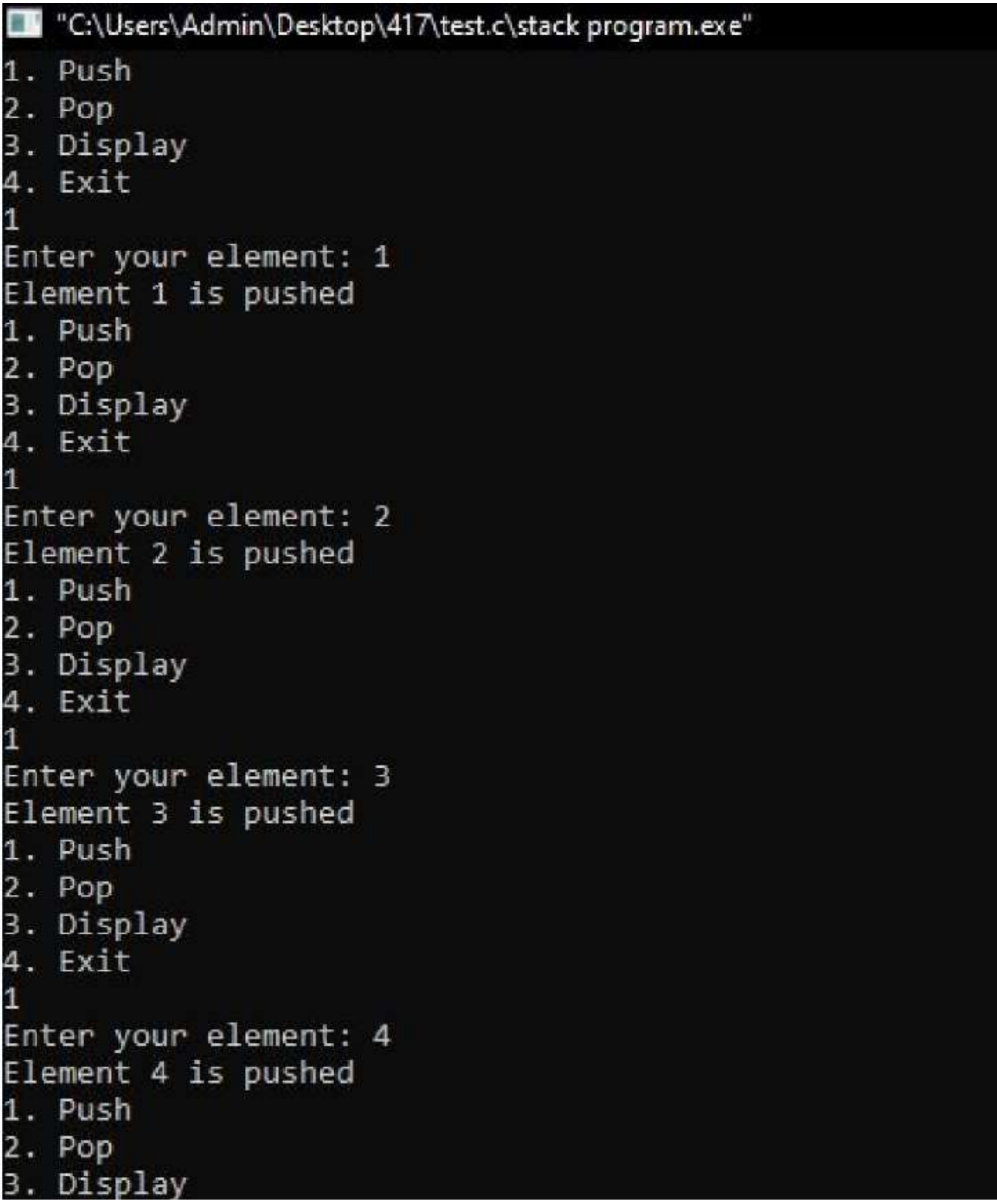
```c
#include <stdio.h>
#include <stdlib.h>
#define size 5

void push(int stack[], int *top, int element) {
    if (*top == size - 1) {
        printf("Stack overflow\n");
        return;
    }
    (*top)++;
    stack[*top] = element;
    printf("Element %d is pushed\n", element);
}

void pop(int stack[], int *top) {
    if (*top == -1) {
        printf("Stack underflow\n");
        return;
    }
    printf("Popped from stack: %d\n", stack[*top]);
    (*top)--;
}

void display(int stack[], int top) {
    if (top == -1) {
        printf("Stack is empty\n");
        return;
    }
    for (int i = top; i >= 0; i--) {
        printf("%d ", stack[i]);
    }
    printf("\n");
}

int main() {
    int stack[size];
    int top = -1;
    int choice, element;

    while (1) {
        printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
        scanf("%d", &choice);
        switch (choice) {
```

```c
void display(int stack[], int top) {
    if (top == -1) {
        printf("Stack is empty\n");
        return;
    }
    for (int i = top; i >= 0; i--) {
        printf("%d ", stack[i]);
    }
    printf("\n");
}

int main() {
    int stack[size];
    int top = -1;
    int choice, element;

    while (1) {
        printf("1. Push\n2. Pop\n3. Display\n4. Exit\n");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter your element: ");
                scanf("%d", &element);
```

```
            scanf("%d", &element);
            push(stack, &top, element);
            break;
        case 2:
            pop(stack, &top);
            break;
        case 3:
            display(stack, top);
            break;
        case 4:
            exit(0);
        default:
            printf("Invalid input\n");
    }
}
}
```


"C:\Users\Admin\Desktop\417\test.c\stack program.exe"
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 1
Element 1 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 2
Element 2 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 3
Element 3 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 4
Element 4 is pushed
1. Push
2. Pop
3. Display

```
Enter your element: 5
Element 5 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 6
Stack overflow
1. Push
2. Pop
3. Display
4. Exit
3
5 4 3 2 1
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 5
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 4
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 3
```

```
Popped from stack: 3
4. Exit
1
Enter your element: 4
Element 4 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 5
Element 5 is pushed
1. Push
2. Pop
3. Display
4. Exit
1
Enter your element: 6
Stack overflow
1. Push
2. Pop
3. Display
4. Exit
3
5 4 3 2 1
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 5
```

```
4. Exit
2
Popped from stack: 5
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 4
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 3
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 2
1. Push
2. Pop
3. Display
4. Exit
2
Popped from stack: 1
1. Push
2. Pop
3. Display
4. Exit
```