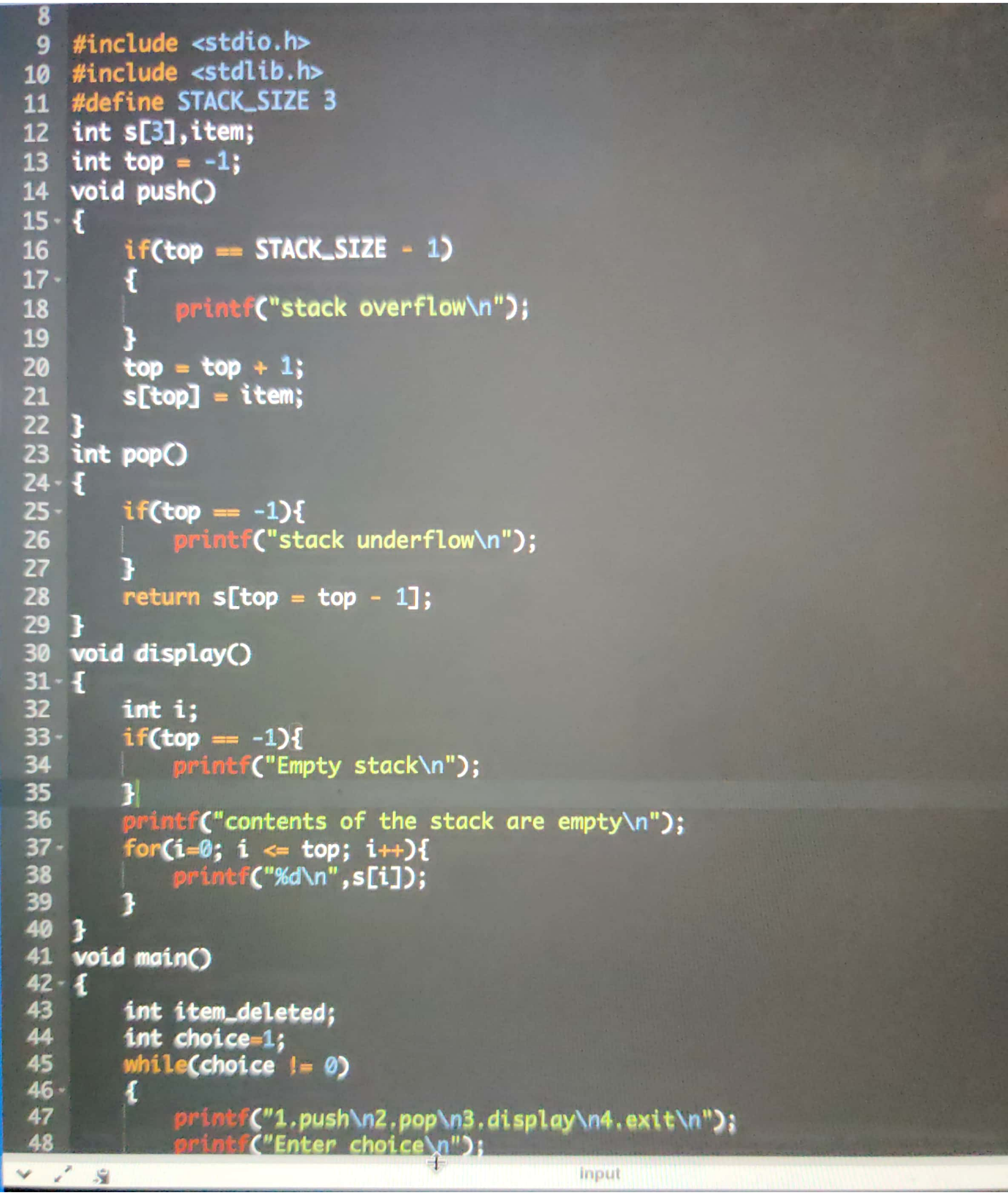
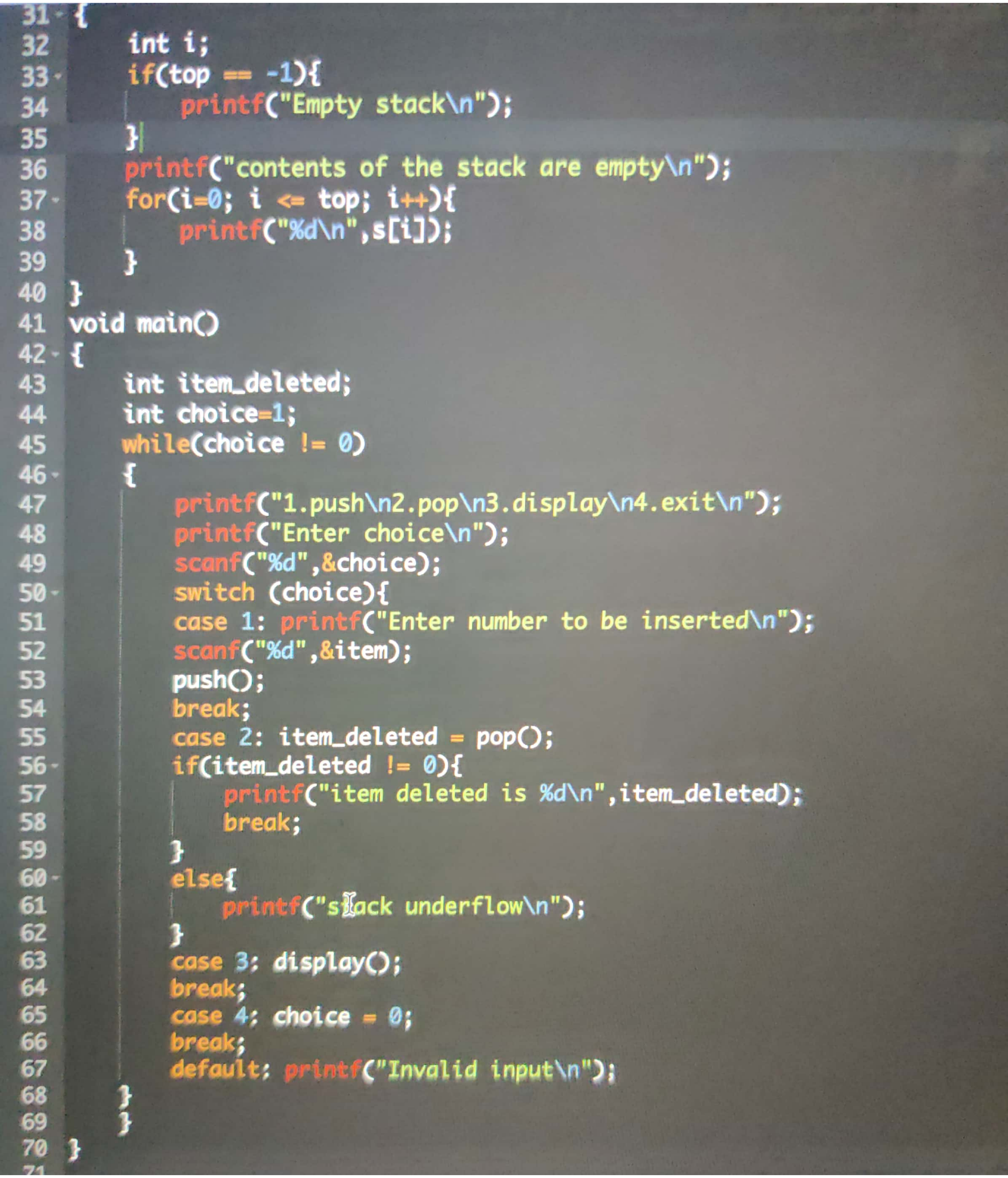
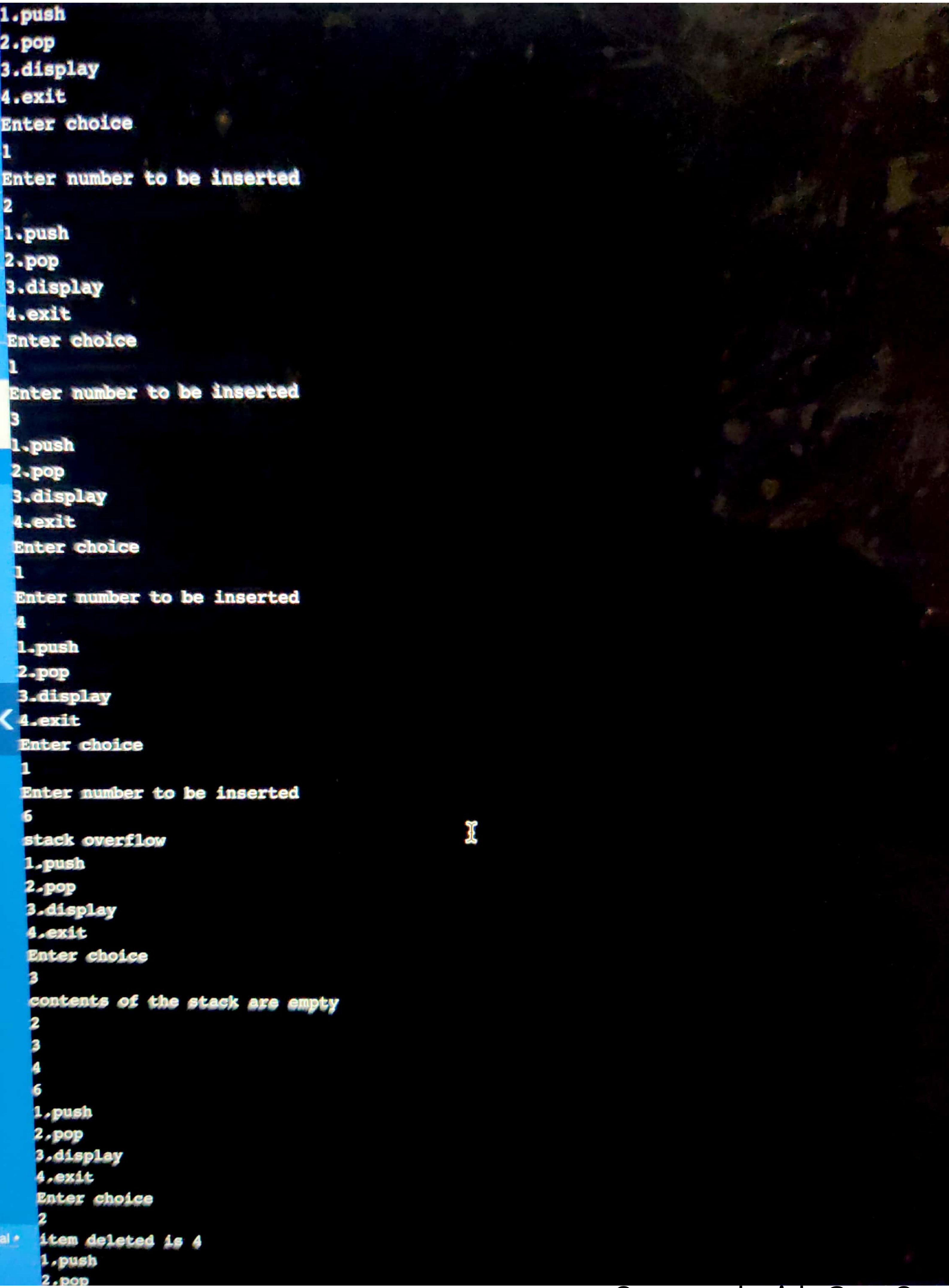


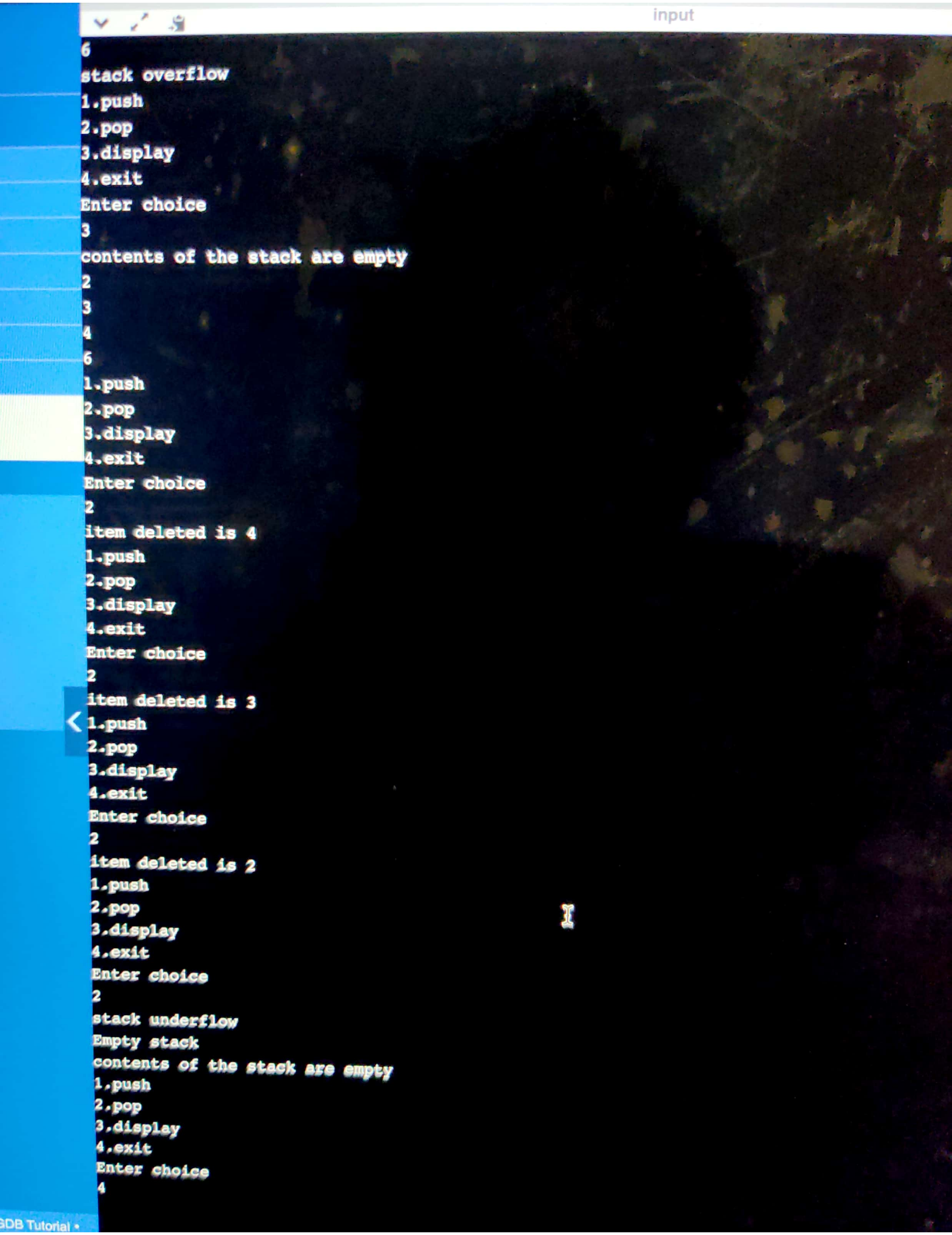
```

#include <stdio.h>
#include <stdlib.h>
#define STACK_SIZE 3
int s[3],item;
int top = -1;
void push()
{
    if(top == STACK_SIZE - 1)
    {
        printf("stack overflow\n");
    }
    top = top + 1;
    s[top] = item;
}
int pop()
{
    if(top == -1){
        printf("stack underflow\n");
    }
    return s[top = top - 1];
}
void display()
{
    int i;
    if(top == -1){
        printf("Empty stack\n");
    }
    printf("contents of the stack are empty\n");
    for(i=0; i <= top; i++){
        printf("%d\n",s[i]);
    }
}
void main()
{
    int item_deleted;
    int choice=1;
    while(choice != 0)
    {
        printf("1.push\n2.pop\n3.display\n4.exit\n");
        printf("Enter choice\n");
```



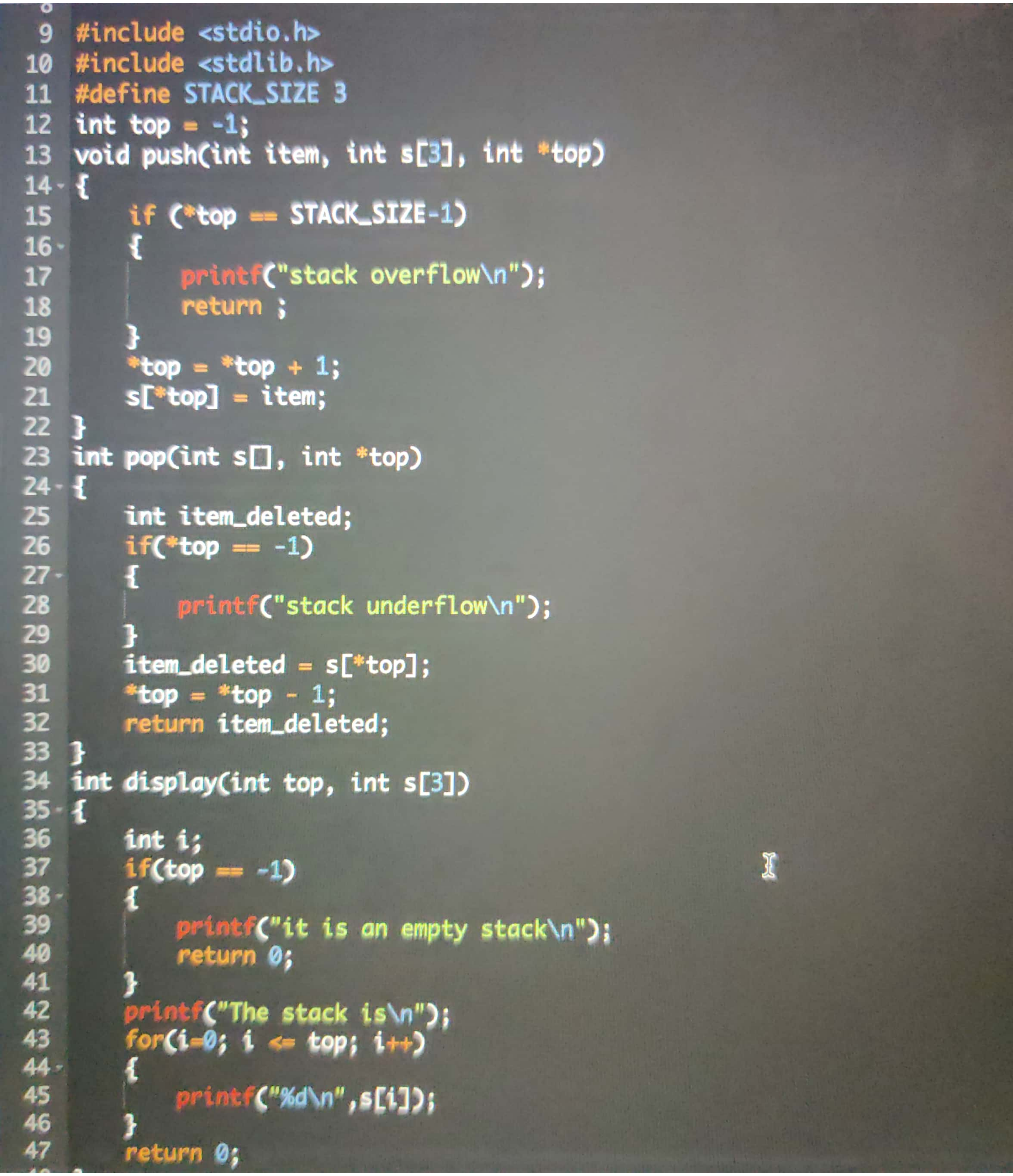
```
{
    int i;
    if(top == -1){
        printf("Empty stack\n");
    }
    printf("contents of the stack are empty\n");
    for(i=0; i <= top; i++){
        printf("%d\n",s[i]);
    }
}
void main()
{
    int item_deleted;
    int choice=1;
    while(choice != 0)
    {
        printf("1.push\n2.pop\n3.display\n4.exit\n");
        printf("Enter choice\n");
        scanf("%d",&choice);
        switch (choice){
        case 1: printf("Enter number to be inserted\n");
        scanf("%d",&item);
        push();
        break;
        case 2: item_deleted = pop();
        if(item_deleted != 0){
            printf("item deleted is %d\n",item_deleted);
            break;
        }
        else{
            printf("stack underflow\n");
        }
        case 3: display();
        break;
        case 4: choice = 0;
        break;
        default: printf("Invalid input\n");
    }
    }
}
```



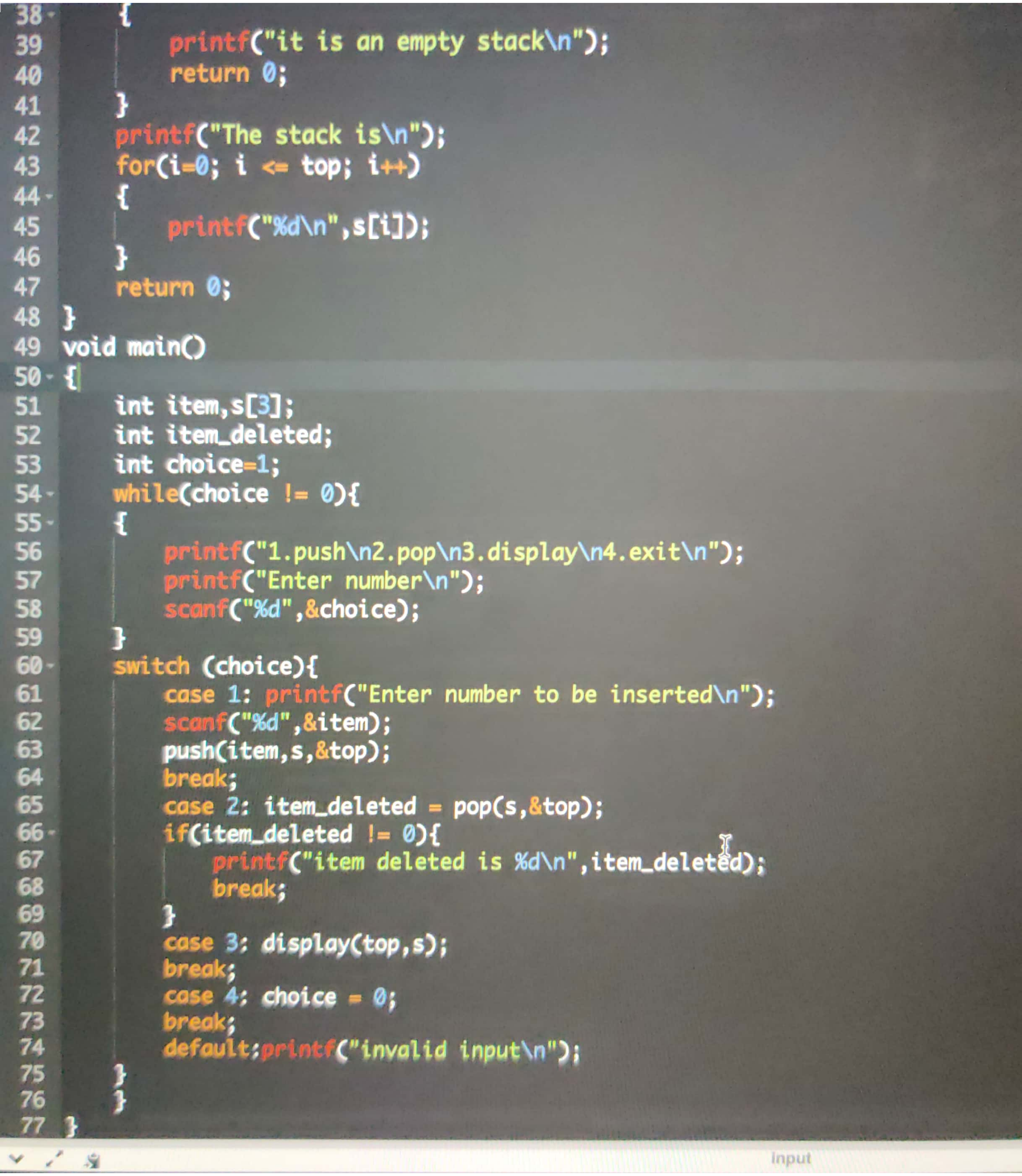
```
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
2
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
3
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
4
1.push
2.pop
3.display
4.exit
Enter choice
1
Enter number to be inserted
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter choice
3
contents of the stack are empty
2
3
4
6
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 4
1.push
2.pop
```



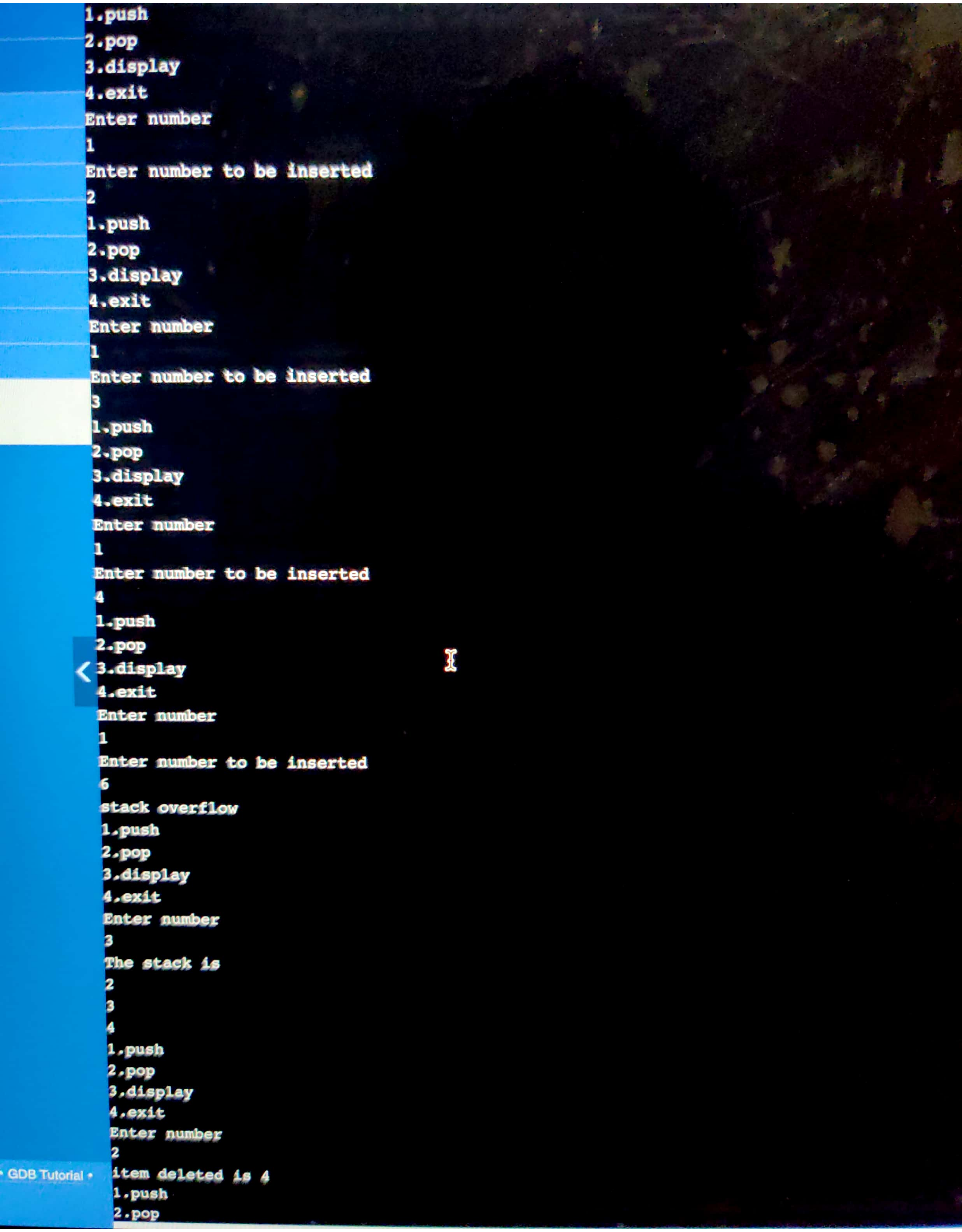
```
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter choice
3
contents of the stack are empty
2
3
4
6
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 4
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 3
1.push
2.pop
3.display
4.exit
Enter choice
2
item deleted is 2
1.push
2.pop
3.display
4.exit
Enter choice
2
stack underflow
Empty stack
contents of the stack are empty
1.push
2.pop
3.display
4.exit
Enter choice
4
```



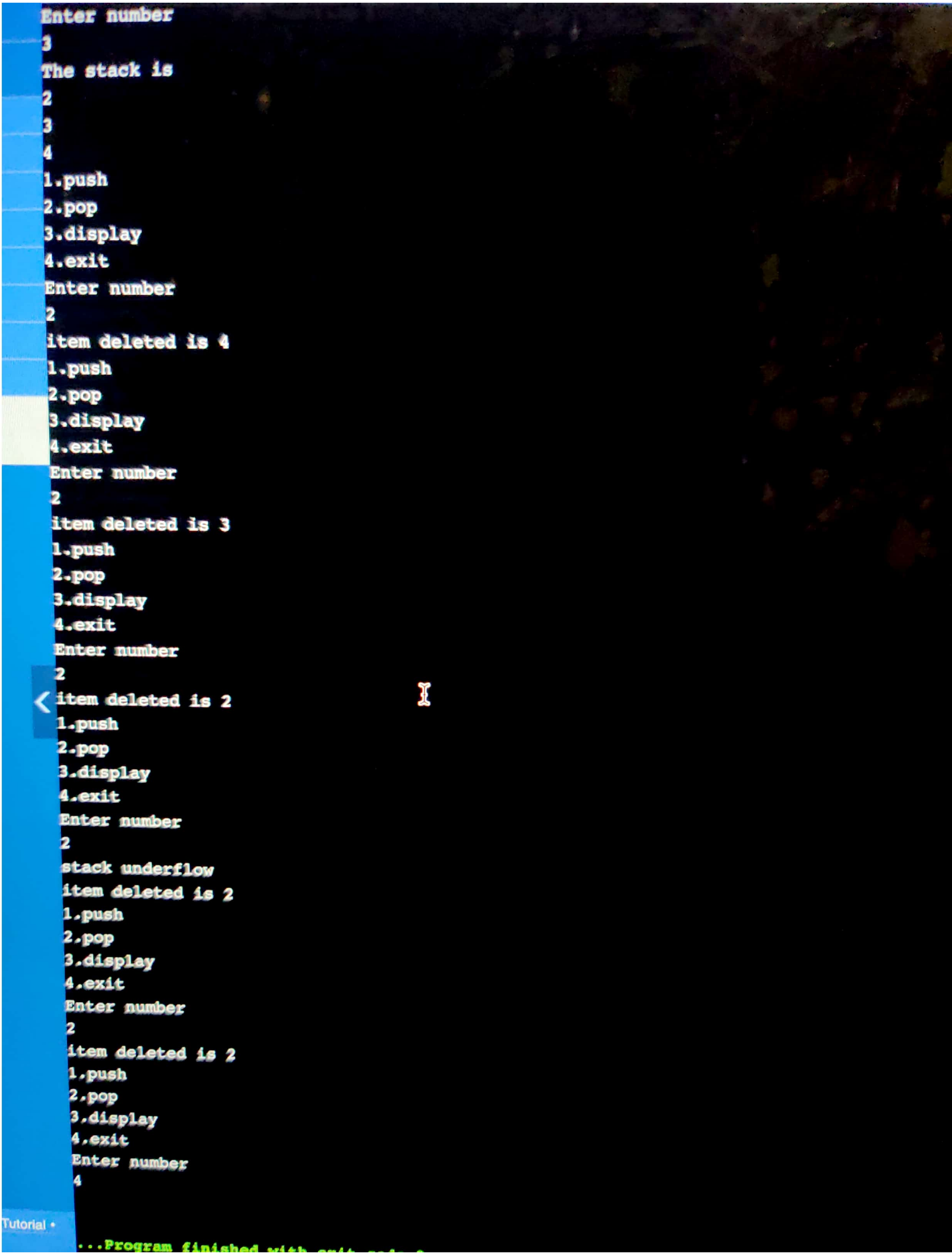
```c
#include <stdio.h>
#include <stdlib.h>
#define STACK_SIZE 3
int top = -1;
void push(int item, int s[3], int *top)
{
    if (*top == STACK_SIZE-1)
    {
        printf("stack overflow\n");
        return ;
    }
    *top = *top + 1;
    s[*top] = item;
}
int pop(int s[], int *top)
{
    int item_deleted;
    if(*top == -1)
    {
        printf("stack underflow\n");
    }
    item_deleted = s[*top];
    *top = *top - 1;
    return item_deleted;
}
int display(int top, int s[3])
{
    int i;
    if(top == -1)
    {
        printf("it is an empty stack\n");
        return 0;
    }
    printf("The stack is\n");
    for(i=0; i <= top; i++)
    {
        printf("%d\n",s[i]);
    }
    return 0;
```



```
    {
        printf("it is an empty stack\n");
        return 0;
    }
    printf("The stack is\n");
    for(i=0; i <= top; i++)
    {
        printf("%d\n",s[i]);
    }
    return 0;
}
void main()
{
    int item,s[3];
    int item_deleted;
    int choice=1;
    while(choice != 0){
    {
        printf("1.push\n2.pop\n3.display\n4.exit\n");
        printf("Enter number\n");
        scanf("%d",&choice);
    }
    switch (choice){
        case 1: printf("Enter number to be inserted\n");
        scanf("%d",&item);
        push(item,s,&top);
        break;
        case 2: item_deleted = pop(s,&top);
        if(item_deleted != 0){
            printf("item deleted is %d\n",item_deleted);
            break;
        }
        case 3: display(top,s);
        break;
        case 4: choice = 0;
        break;
        default:printf("invalid input\n");
    }
    }
}
```



```
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
2
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
3
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
4
1.push
2.pop
3.display
4.exit
Enter number
1
Enter number to be inserted
6
stack overflow
1.push
2.pop
3.display
4.exit
Enter number
3
The stack is
2
3
4
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 4
1.push
2.pop
```



```
Enter number
3
The stack is
2
3
4
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 4
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 3
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 2
1.push
2.pop
3.display
4.exit
Enter number
2
stack underflow
item deleted is 2
1.push
2.pop
3.display
4.exit
Enter number
2
item deleted is 2
1.push
2.pop
3.display
4.exit
Enter number
4


...Program finished with exit code 0
```