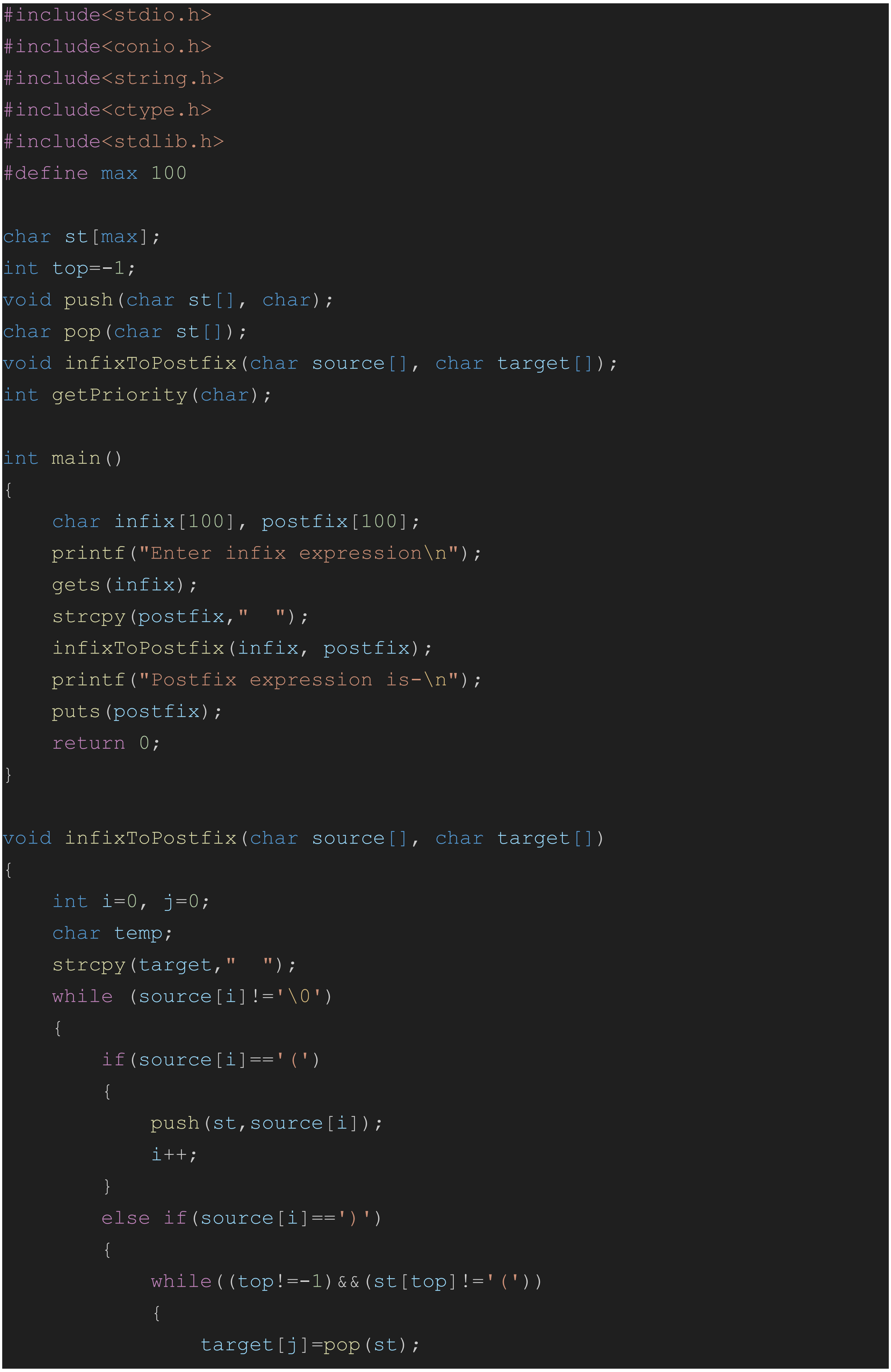
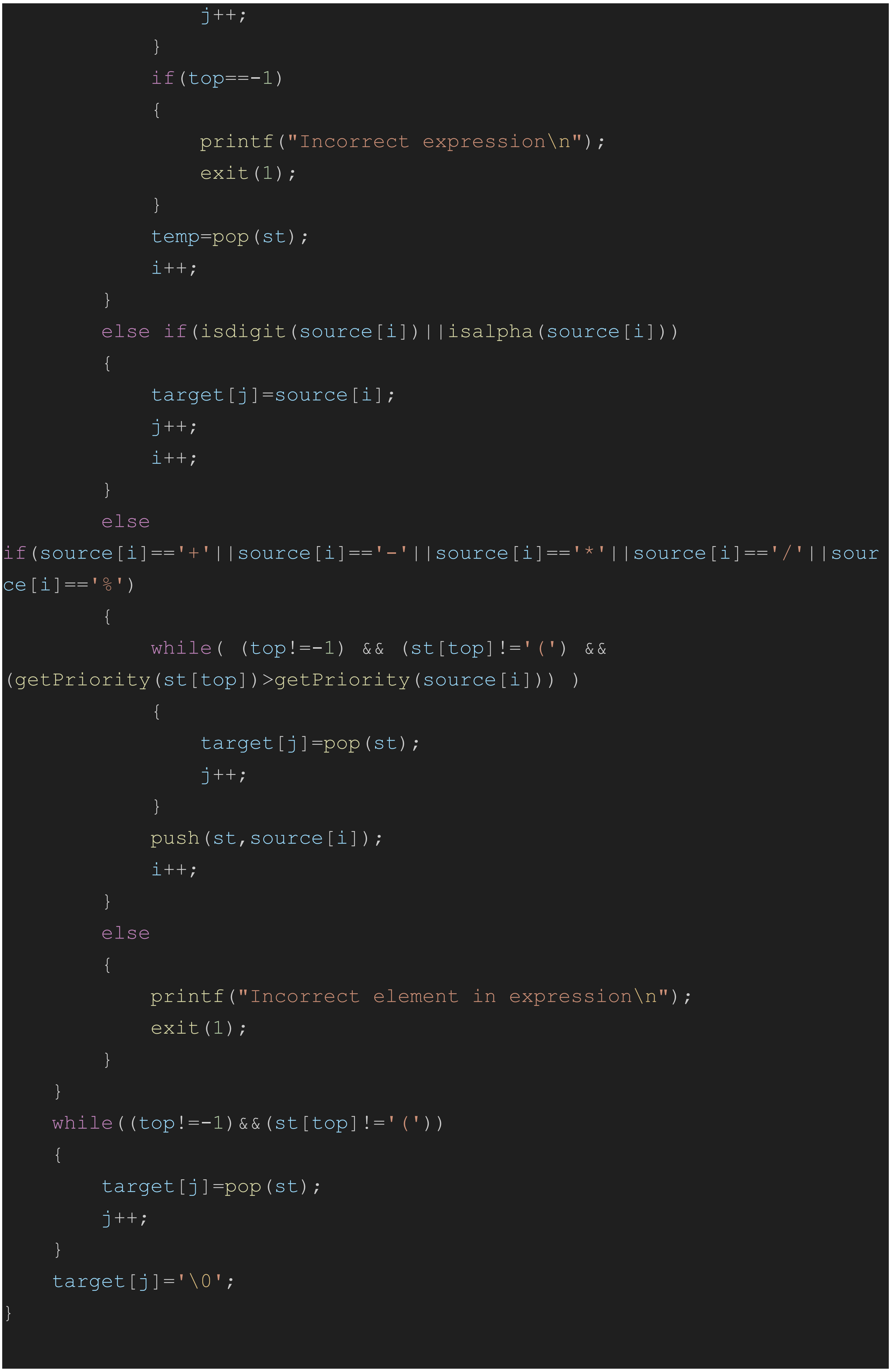
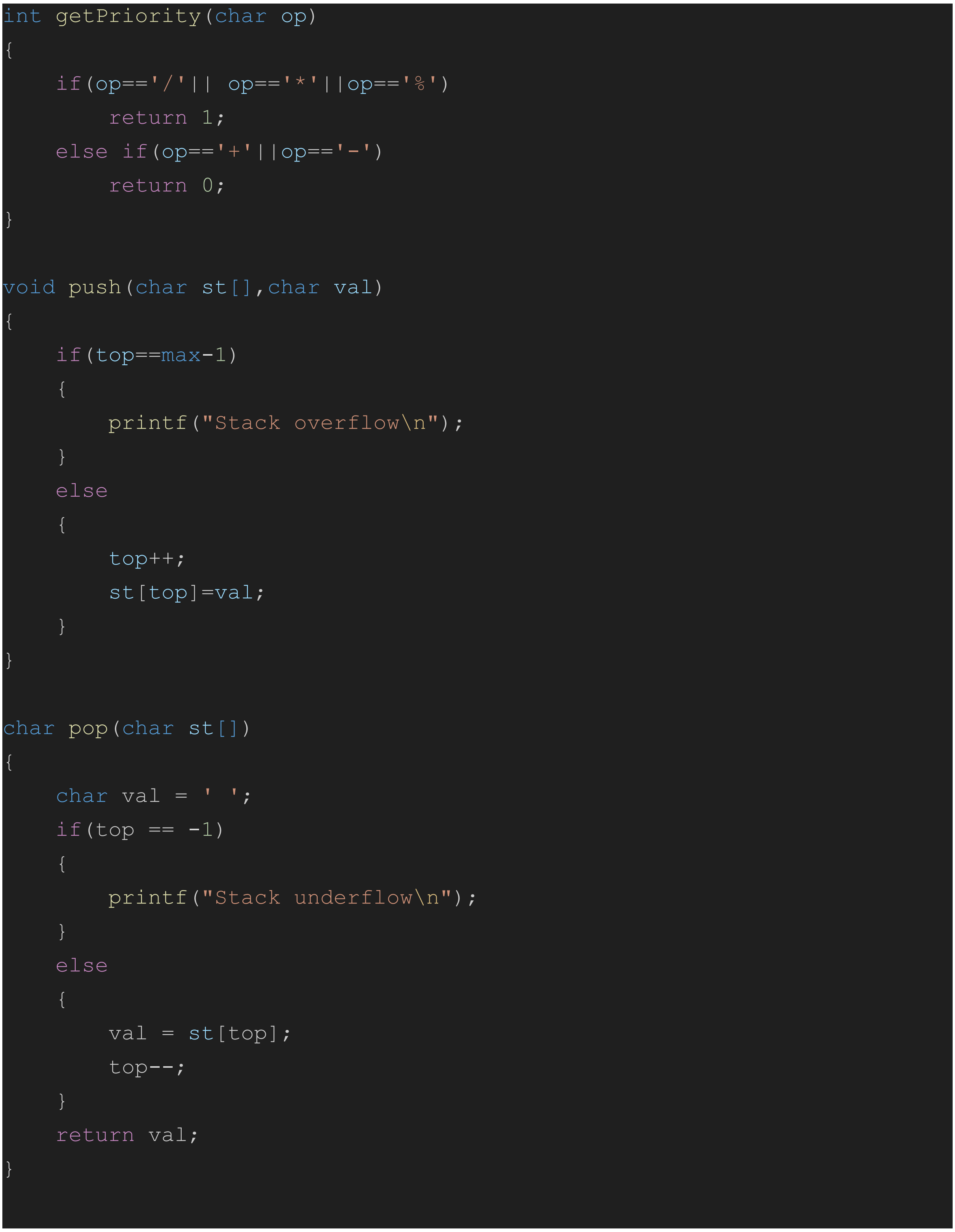


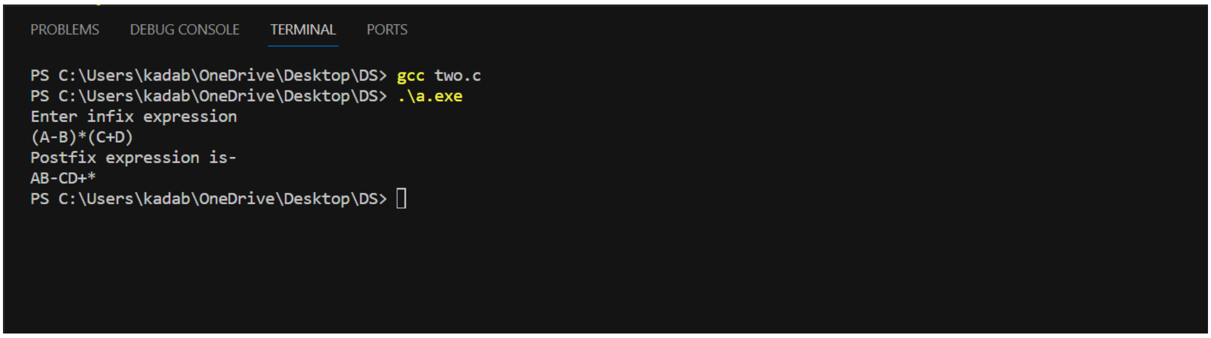
```c
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#define max 100

char st[max];
int top=-1;
void push(char st[], char);
char pop(char st[]);
void infixToPostfix(char source[], char target[]);
int getPriority(char);

int main()
{
    char infix[100], postfix[100];
    printf("Enter infix expression\n");
    gets(infix);
    strcpy(postfix,"  ");
    infixToPostfix(infix, postfix);
    printf("Postfix expression is-\n");
    puts(postfix);
    return 0;
}

void infixToPostfix(char source[], char target[])
{
    int i=0, j=0;
    char temp;
    strcpy(target,"  ");
    while (source[i]!='\0')
    {
        if(source[i]=='(')
        {
            push(st,source[i]);
            i++;
        }
        else if(source[i]==')')
        {
            while((top!=-1)&&(st[top]!='('))
            {
                target[j]=pop(st);
```



```
                j++;
            }
            if(top==-1)
            {
                printf("Incorrect expression\n");
                exit(1);
            }
            temp=pop(st);
            i++;
        }
        else if(isdigit(source[i])||isalpha(source[i]))
        {
            target[j]=source[i];
            j++;
            i++;
        }
        else if(source[i]=='+'||source[i]=='-'||source[i]=='*'||source[i]=='/'||source[i]=='%')
        {
            while( (top!=-1) && (st[top]!='(') && (getPriority(st[top])>getPriority(source[i])) )
            {
                target[j]=pop(st);
                j++;
            }
            push(st,source[i]);
            i++;
        }
        else
        {
            printf("Incorrect element in expression\n");
            exit(1);
        }
    }
    while((top!=-1)&&(st[top]!='('))
    {
        target[j]=pop(st);
        j++;
    }
    target[j]='\0';
}
```



```
int getPriority(char op)
{
    if(op=='/'|| op=='*'||op=='%')
        return 1;
    else if(op=='+'||op=='-')
        return 0;
}

void push(char st[],char val)
{
    if(top==max-1)
    {
        printf("Stack overflow\n");
    }
    else
    {
        top++;
        st[top]=val;
    }
}

char pop(char st[])
{
    char val = ' ';
    if(top == -1)
    {
        printf("Stack underflow\n");
    }
    else
    {
        val = st[top];
        top--;
    }
    return val;
}
```



```
PS C:\Users\kadab\OneDrive\Desktop\DS> gcc two.c
PS C:\Users\kadab\OneDrive\Desktop\DS> .\a.exe
Enter infix expression
(A-B)*(C+D)
Postfix expression is-
AB-CD+*
PS C:\Users\kadab\OneDrive\Desktop\DS> 
```

1/1/2024