LAB PROGRAM 2: 30/09/2020

main.cpp ✕ main.c ✕

```c
     1    #include <stdio.h>
     2    #include <stdlib.h>
     3    #include <string.h>
     4    #include <process.h>
     5    int F(char symbol)
     6    {
     7        switch(symbol)
     8        {
     9        case '+':
    10        case '-':return 2;
    11        case '*':
    12        case '/':return 4;
    13        case '^':
    14        case '$':return 5;
    15        case '(':return 0;
    16        case '#':return -1;
    17        default:return 8;
    18        }
    19
    20    }
    21    int G(char symbol)
    22    {
    23        switch(symbol)
    24        {
    25        case '+':
    26        case '-':return 1;
```
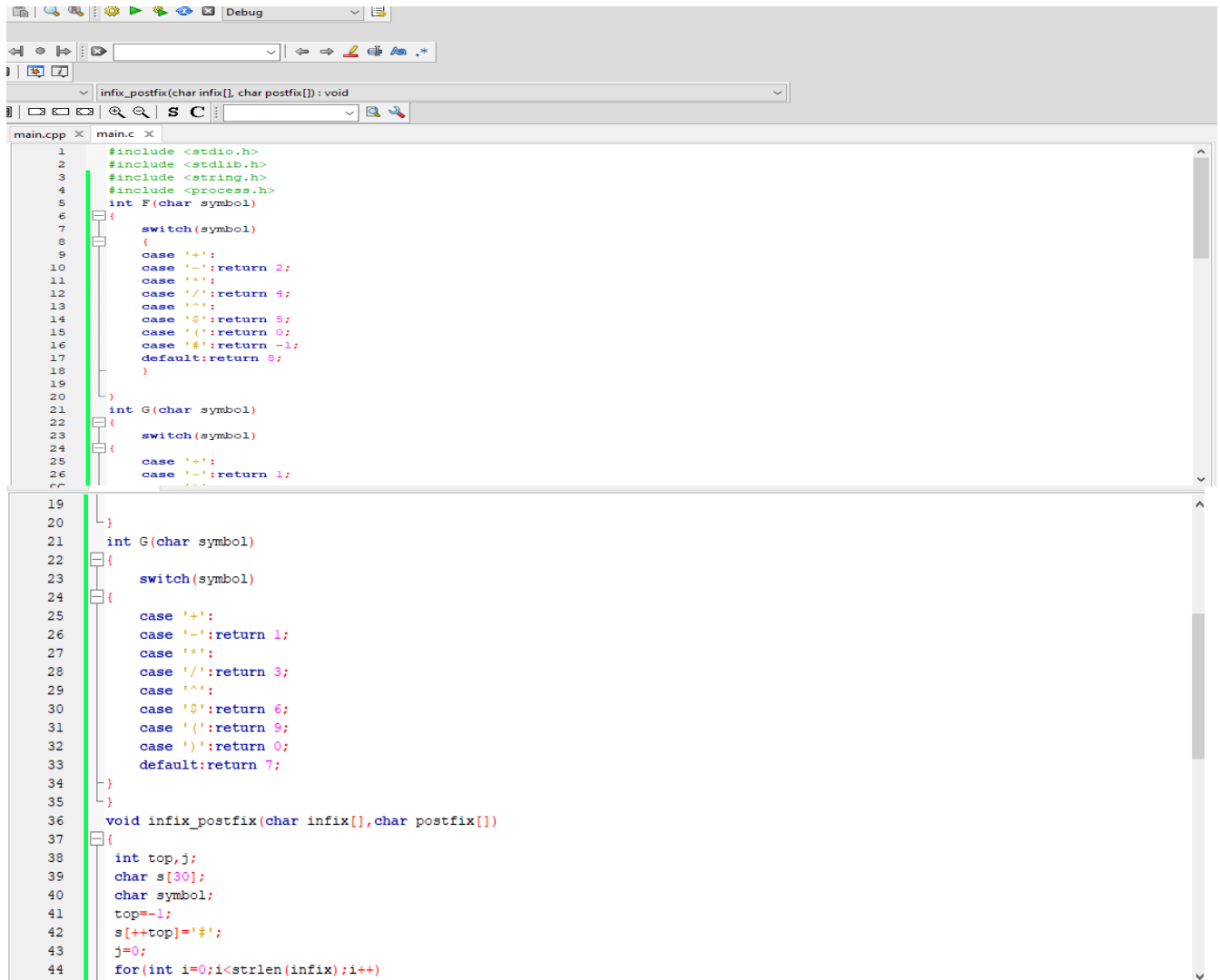
```c
    19
    20    }
    21    int G(char symbol)
    22    {
    23        switch(symbol)
    24        {
    25        case '+':
    26        case '-':return 1;
    27        case '*':
    28        case '/':return 3;
    29        case '^':
    30        case '$':return 6;
    31        case '(':return 9;
    32        case ')':return 0;
    33        default:return 7;
    34    }
    35    }
    36    void infix_postfix(char infix[],char postfix[])
    37    {
    38      int top,j;
    39      char s[30];
    40      char symbol;
    41      top=-1;
    42      s[++top]='#';
    43      j=0;
    44      for(int i=0;i<strlen(infix);i++)
```

```c
        j=0;
        for(int  i=0;i<strlen(infix);i++)
        {
            symbol=infix[i];
            while(F(s[top])>G(symbol))
            {
                postfix[j]=s[top--];
                j++;
            }
            if(F(s[top])!=G(symbol))
            {
                    s[++top]=symbol;
            }

            else
        {
            top--;
        }

        }
    while(s[top]!='#')
    {
    postfix[j++]=s[top--];
    }
    postfix[j]='\0';
    }
```

```c
        else
    {
        top--;
    }

    }
    while(s[top]!='#')
    {
    postfix[j++]=s[top--];
    }
    postfix[j]='\0';
    }
    int main()
    {
        char infix[20];
        char postfix[20];
        printf("Please enter the valid infix expression\n");
        scanf("%s",infix);
        infix_postfix(infix,postfix);
     printf("the postfix is \n");
    printf("%s",postfix);

        return 0;
    }
```

```
Please enter the valid infix expression
(A+(B-C)*D)
the postfix is
ABC-D*+
Process returned 0 (0x0)   execution time : 7.400 s
Press any key to continue.
```

```
Please enter the valid infix expression
a+b*(c^d-e)^(f+g*h)-i
the postfix is
abcd^e-fgh*+^*+i-
Process returned 0 (0x0)   execution time : 2.890 s
Press any key to continue.
```

```
Please enter the valid infix expression
X^Y^Z-M+N+P/Q
the postfix is
XYZ^^M-N+PQ/+
Process returned 0 (0x0)   execution time : 1.739 s
Press any key to continue.
```