```
Run >
main.c
                                                                                       1
      #include<stdio.h>
  2
      #include<string.h>
  3
      #include<stdlib.h>
  4
  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'$':return 5;
 15
       case'(':return 0;
       case'#':return -1;
 16
 17
       default:return 8;
 18
       }
 19
                                                                                     I
 20
 21
       int G(char symbol)
 22
       {
       switch(symbol)
 23
 24
        {
 25
       case'+':
 26
       case'-':return 1;
        case'*':
 27
 28
        case'/':return 3;
       case'^':
 29
       case'$':return 6;
 30
       case'(':return 9;
 31
        case')':return 0;
 32
 33
        default: return 7;
 34
       }
 35
       }
 36
 37
      void infix_postfix(char infix[],char postfix[])
 38
       {
 39
      int top, i, j;
      char s[30],symbol;
 40
 41
      top=-1;
 42
       s[++top]='#';
 43
      j=0;
 44
 45
       for(i=0;i<strlen(infix);i++)</pre>
```

```
et C v 🕤
                                                                      Run >
  main.c
                                                                                            42
         s[++top]='#';
    43
         j=0;
    44
         for(i=0;i<strlen(infix);i++)</pre>
    45
    46
          {
         symbol=infix[i];
    47
    48
         while(F(s[top])>G(symbol))
    49
          {
          postfix[j]=s[top--];
    50
    51
          j++;
    52
          }
    53
     54
     55
          if(F(s[top])!=G(symbol))
     56
          s[++top]=symbol;
     57
          else
                                                                    I
     58
          top--;
     59
          }
     60
     61
          while(s[top]!='#')
     62
           {
     63
          postfix[j++]=s[top--];
     64
     65
           postfix[j]='\0';
     66
      67
      68
           int main()
      69
      70
           char infix[20];
      71
           char postfix[20];
           printf("Enter the valid infix expression ");
      72
      73
           scanf("%s", infix);
           infix_postfix(infix , postfix );
      74
           printf("The postfix expression is \n");
      75
      76
           printf("%s\n", postfix);
      77
```

```
clang-7 -pthread -lm -o main main.c

./main

Enter the valid infix expression (1+2)*(3/4)*(6-2)

The postfix expression is

12+34/*62-*

. []
```

```
clang-7 -pthread -lm -o main main.c

./main

Enter the valid infix expression exited, terminated

(A+B)*(C/D

bash: syntax error near unexpected token `*'

(A+B)(C/D

bash: syntax error near unexpected token `('

[]
```

S.R.P00JA 1BM19CS135	classmate
CSE	Date
DS LAB	
#include < stdio.h>	7.
# include (string . h >	::
# welude < stallib.h >	Jalouel 17572 poly
int F (char symbol)	L- = 00T
t	"H" 1 057 0+ 12
Switch (Symbol)	0.22
3	(m) (ma) wide = 1 :0 = 1 3ak
cage '+'!	3
case '-': setuen 2;	Witness Statement
case '#':	(Colonie) D < (Eggs) 2 19) etilar
case 11': setuen 4;	7
case 'N':	C-anda e (1) more
case '\$' : seturn 5;	++2
case 'C': setuen o'	
case '#' : seturn -1'	((John vie) Do 1 (Toro) A) Is
default : setuen 8;	Jahre Tartes 8
3	eta
3	(20
int Gr (char symbol)	1
{	Capt = 1 (god) e lister
Switch (Symbol)	3
£	peller [gre] a Stop];
case '+':	1
case'-' ! Return 1',	policipo o vo:
can '* :	2
case '/' = settur 3;	Chiero to
case 'A':	3
case '\$' = Return 6;	italingi 2011
case ('s: getuen 9;	
(ase) &: gettin 0;	with Index with while I string
default: letuer 7;	conflue a prophose
3	(Capter man (super police)
3 " (ON) A	and (" Be nother consisting

```
int top, ij,
char & [30], symbol;
top = -1;
&[++top]= `#';
for ( i=0; i < stalen (injon); i++)
symbol= infox[i];
while (F(8[top])>G(symbol))
postfix [j]= s[top--];
if (F(s [top])! = G(symbol))
S[++top]= symbol;
while (8[top]! = '#')
postfre [j+t] = 8 [top--];
int main ()
char inforcio);
Char postfix [20],
painty ("Enter the valid infox expression");
scanf ("/ 2, infor);
infox - postfox (sinfox, postfox)
pront (" The postfix expression is \n");
print ("1.8\n", postfix);
```