Lab Program 2:-

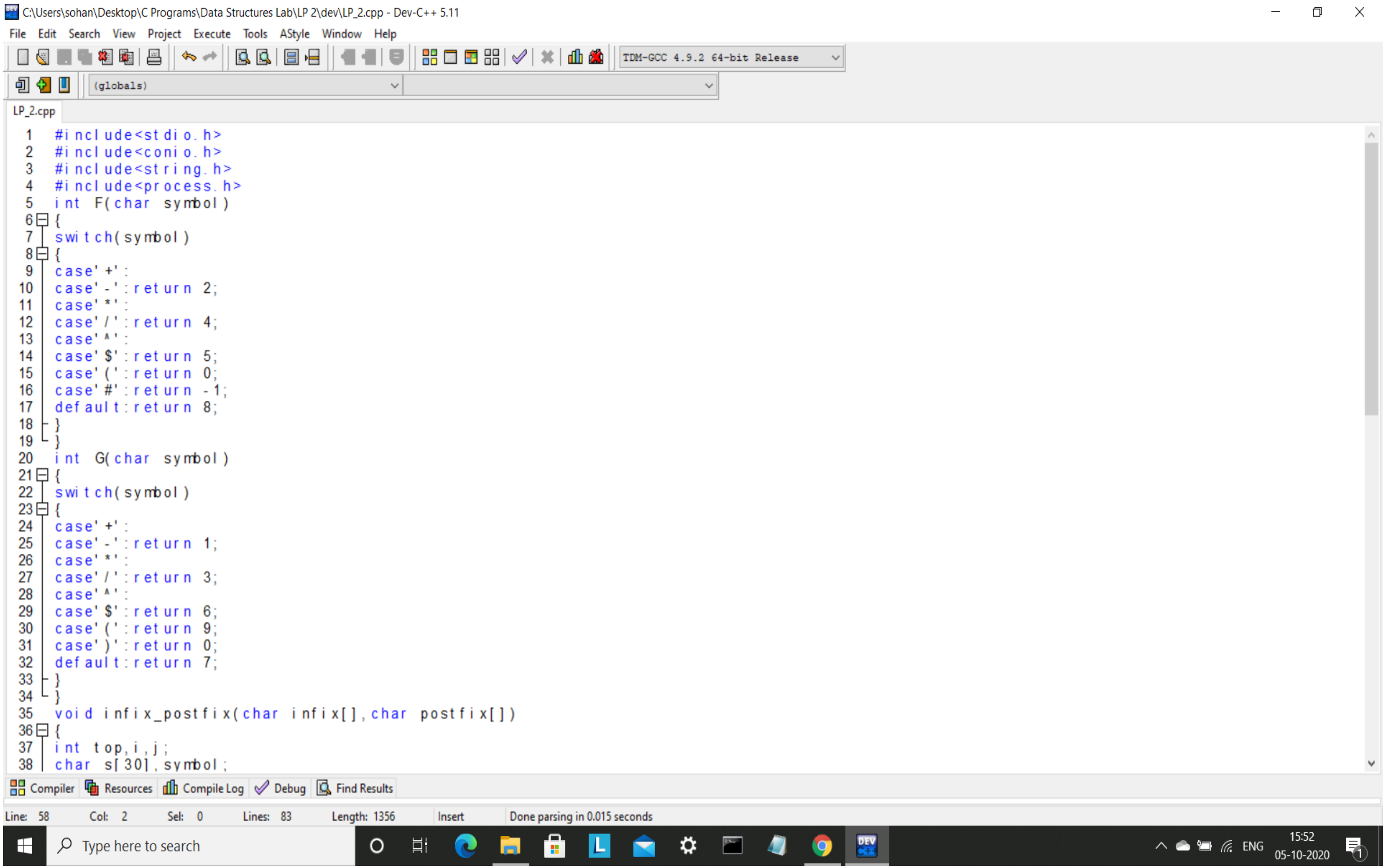
Q} Write a program a given valid parenthesized infix arithmetic expression to postfix expression. The expression consists of single character operands and binary operators + (plus), - (minus), * (multiply) & / (divide).

A.
```
#include<stdio.h>
#include< conio.h>
#include< string.h>
#include< process.h>
int F(char symbol>
{ switch ( symbol )
 { case '+':
   case '-': return 2;
   case '*':
   case '/': return 4;
   case '^':
   case '$' : return 5;
   case '(': return 0;
   case '#': return -1;
   default : return 8;
}}
 int G(char symbol)
{ switch(symbol)
{  case '+':
   case '-': return 1;
   case '*':
   case '/': return 3;
   case '^';
   case '$': return 6;
   case '(': return 9;
   case ')': return 0;
   default: return 7;
}}
```
P.T.O.

```
void infix_postfix(char infix[], char postfix[])
{
 int top, i, j;
 char s[30], symbol;
 top = -1;
s[++top] = '#';
j=0;
for(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';
}
int main()
{ int t;
 char infix[20];
 char postfix[20];
 system("cls");
 printf("Enter the valid infix expression \n");
Scanf("%s", infix);
for(t=0; t< strlen(infix); t++) {
```

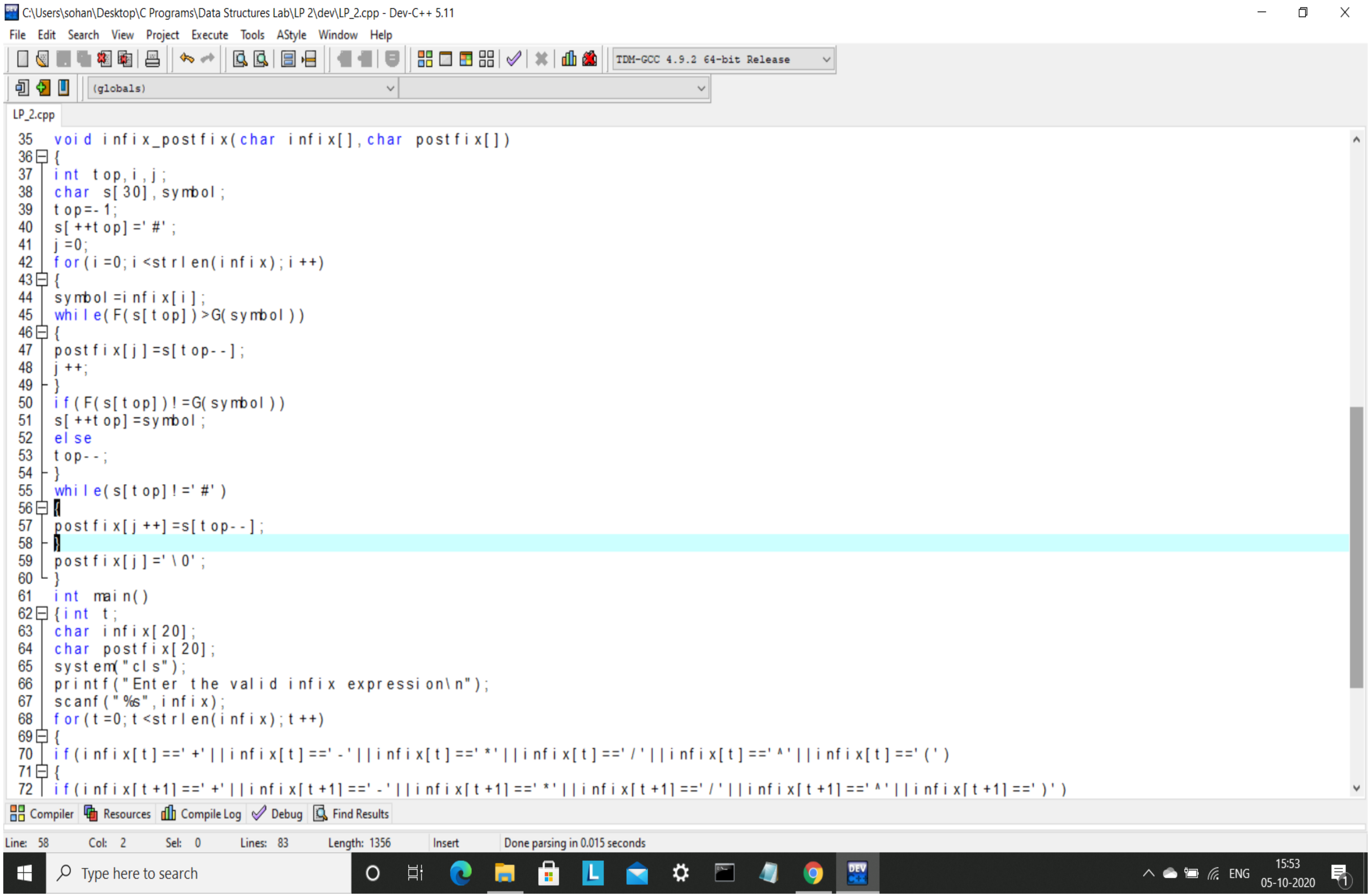
```
if(infix[t]=='+' || infix[t]=='-'|| infix[t]=='*' ||
 infix[t]=='/'|| infix[t]=='^'|| infix[t]=='(')
{
if(infix[t+1]=='+'|| infix[t+1]=='-' || infix[t+1]=='*'
 || infix[t+1]=='/'|| infix[t+1]=='^' || infix[t+1]==')');
{
 printf("Invalid"); exit(0);
}}}

infix_postfix(infix, postfix);
printf("the postfix exp is \n");
getch();
return 0;
}
```


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\LP 2\dev\LP_2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
LP_2.cpp
Compiler Resources Compile Log Debug Find Results
Type here to search
ENG 15:52 05-10-2020

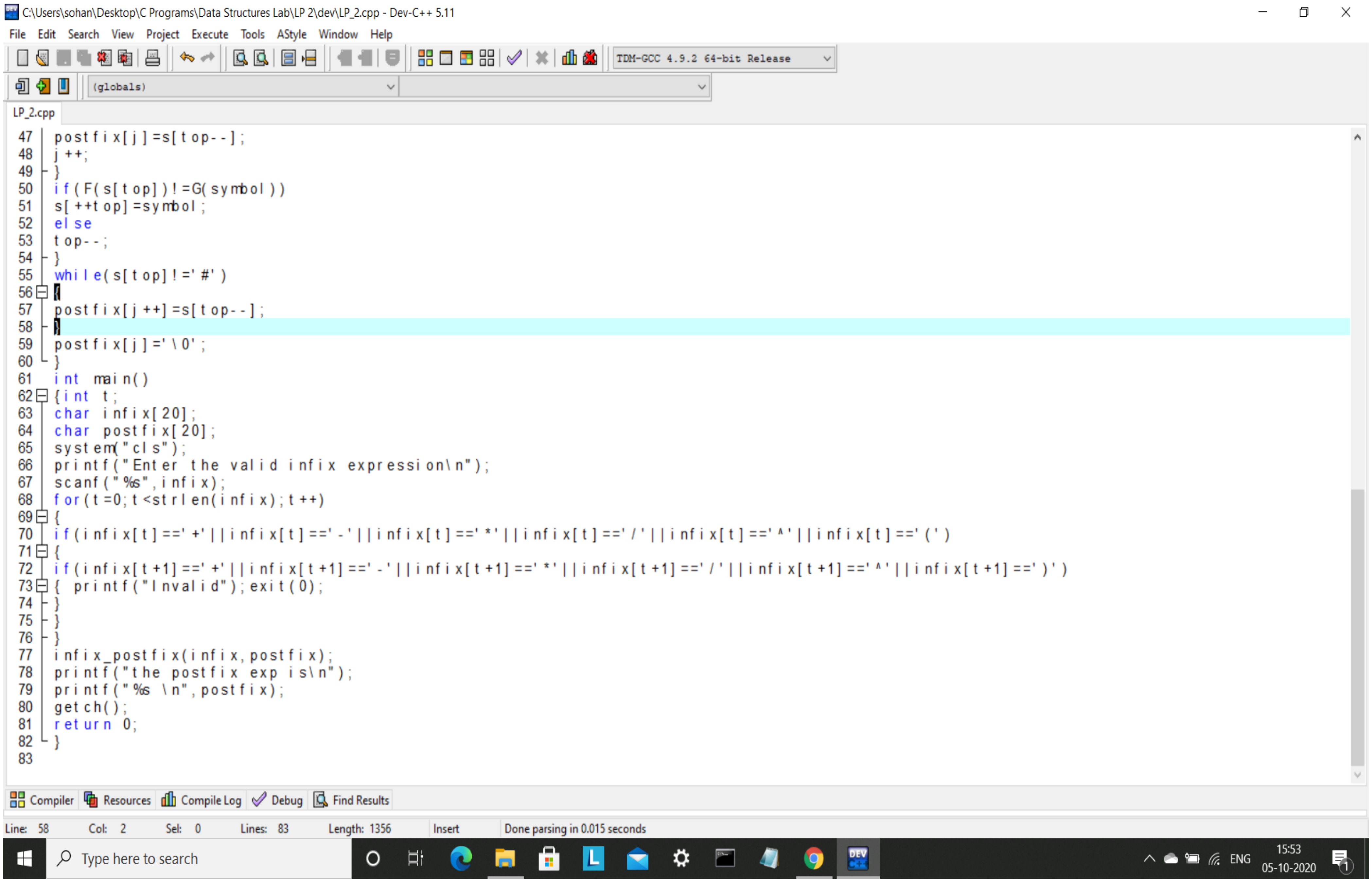
```
#include<stdio.h>
#include<conio.h>
#include<string.h>
#include<process.h>
int F(char symbol)
{
switch(symbol)
{
case'+':
case'-':return 2;
case'*':
case'/':return 4;
case'^':
case'$':return 5;
case'(':return 0;
case'#':return -1;
default:return 8;
}
}
int G(char symbol)
{
switch(symbol)
{
case'+':
case'-':return 1;
case'*':
case'/':return 3;
case'^':
case'$':return 6;
case'(':return 9;
case')':return 0;
default:return 7;
}
}
void infix_postfix(char infix[],char postfix[])
{
int top,i,j;
char s[30],symbol;
```

Line: 58 Col: 2 Sel: 0 Lines: 83 Length: 1356 Insert Done parsing in 0.015 seconds


C:\Users\sohan\Desktop\C Programs\Data Structures Lab\LP 2\dev\LP_2.cpp - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
LP_2.cpp

```c
void infix_postfix(char infix[],char postfix[])
{
int top,i,j;
char s[30],symbol;
top=-1;
s[++top]='#';
j=0;
for(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';
}
int main()
{int t;
char infix[20];
char postfix[20];
system("cls");
printf("Enter the valid infix expression\n");
scanf("%s",infix);
for(t=0;t<strlen(infix);t++)
{
if(infix[t]=='+'||infix[t]=='-'||infix[t]=='*'||infix[t]=='/'||infix[t]=='^'||infix[t]=='(')
{
if(infix[t+1]=='+'||infix[t+1]=='-'||infix[t+1]=='*'||infix[t+1]=='/'||infix[t+1]=='^'||infix[t+1]==')')
```

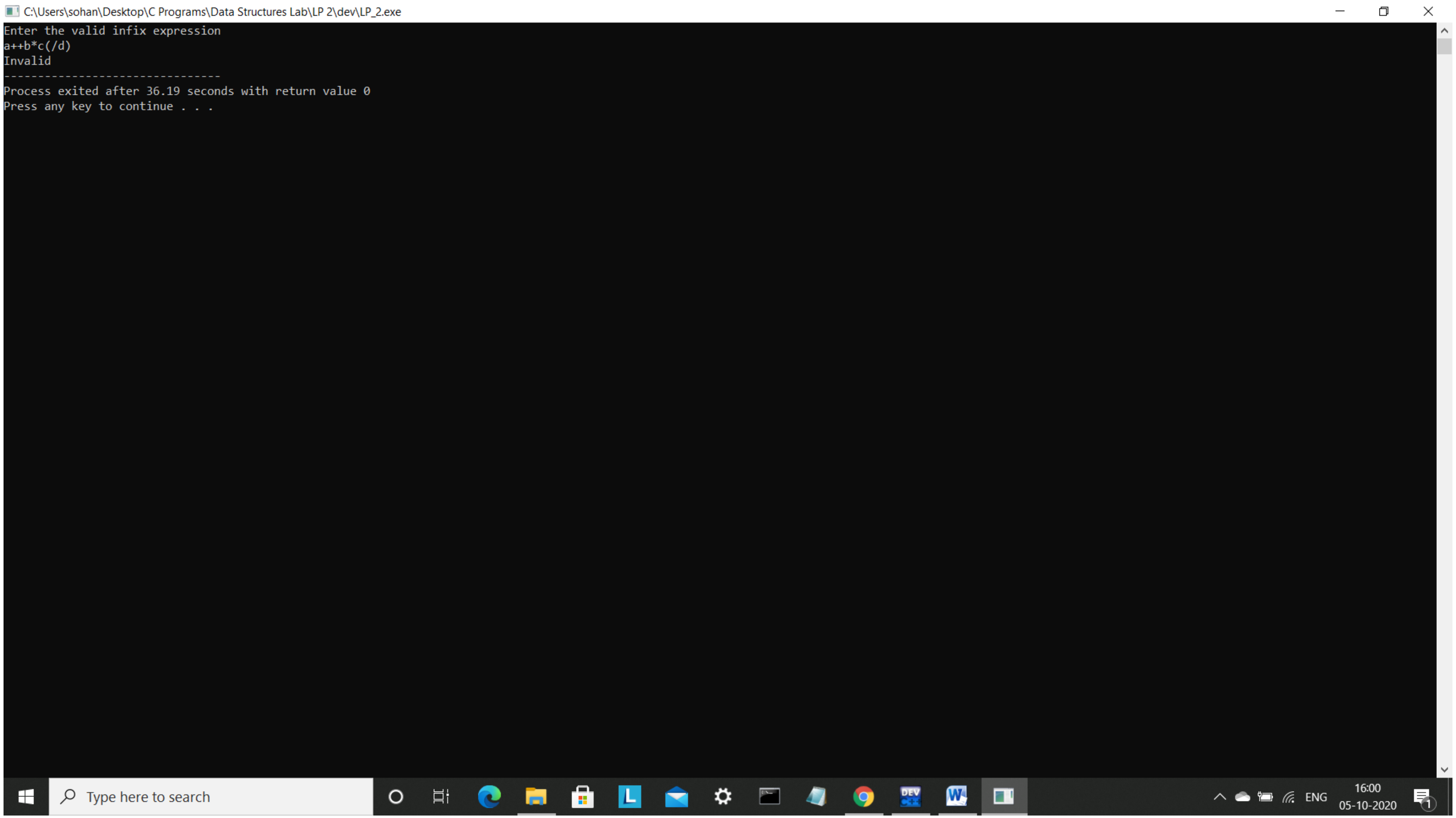
```
Line: 58    Col: 2    Sel: 0    Lines: 83    Length: 1356    Insert    Done parsing in 0.015 seconds
```



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\LP 2\dev\LP_2.cpp - Dev-C++ 5.11

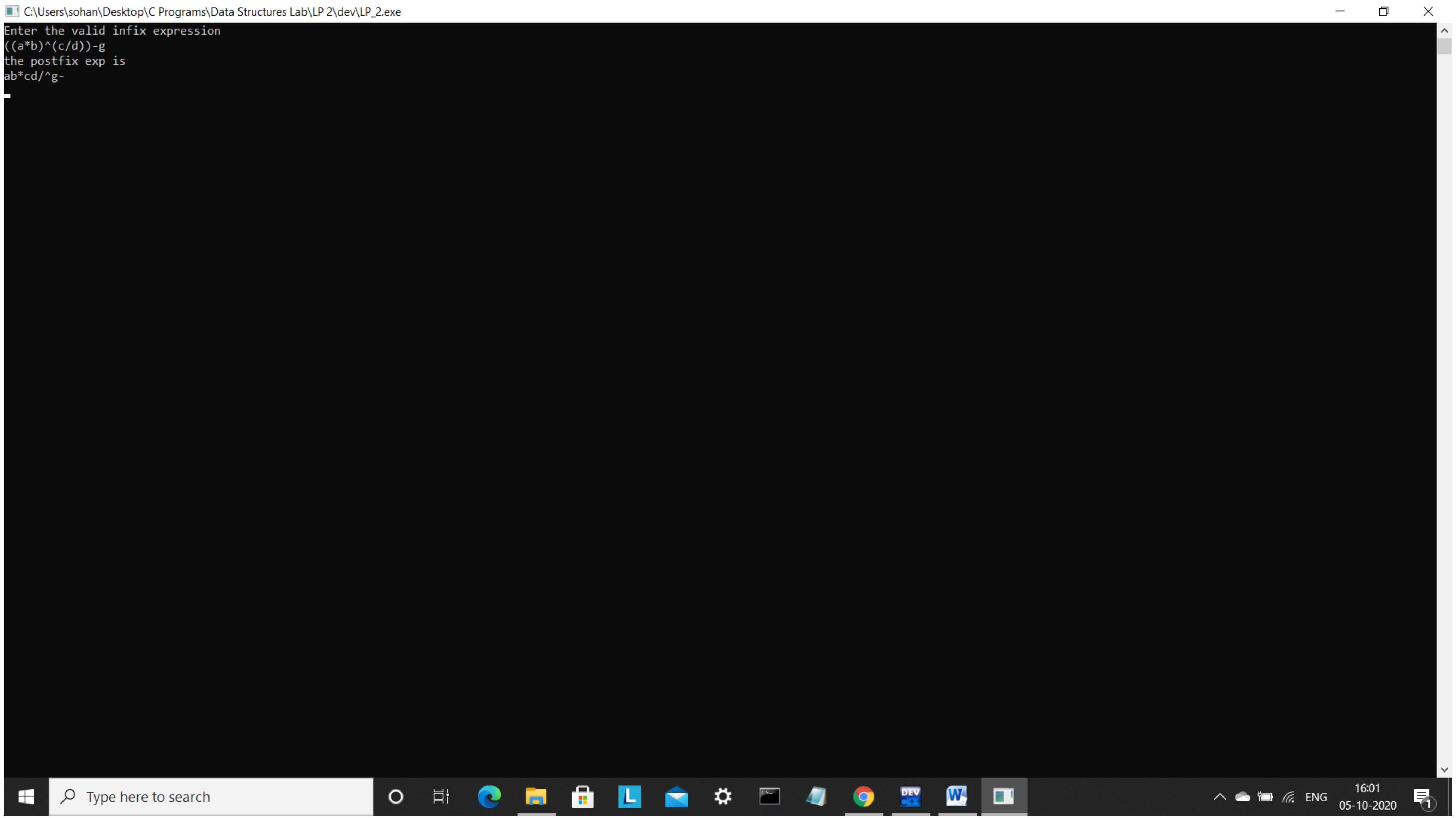
```
postfix[j]=s[top--];
j++;
}
if(F(s[top])!=G(symbol))
s[++top]=symbol;
else
top--;
}
while(s[top]!='#')
{
postfix[j++]=s[top--];
}
postfix[j]='\0';
}
int main()
{int t;
char infix[20];
char postfix[20];
system("cls");
printf("Enter the valid infix expression\n");
scanf("%s",infix);
for(t=0;t<strlen(infix);t++)
{
if(infix[t]=='+'||infix[t]=='-'||infix[t]=='*'||infix[t]=='/'||infix[t]=='^'||infix[t]=='(')
{
if(infix[t+1]=='+'||infix[t+1]=='-'||infix[t+1]=='*'||infix[t+1]=='/'||infix[t+1]=='^'||infix[t+1]==')')
{ printf("Invalid");exit(0);
}
}
}
infix_postfix(infix,postfix);
printf("the postfix exp is\n");
printf("%s \n",postfix);
getch();
return 0;
}
```

File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
LP_2.cpp
Compiler Resources Compile Log Debug Find Results
Line: 58 Col: 2 Sel: 0 Lines: 83 Length: 1356 Insert Done parsing in 0.015 seconds
Type here to search
ENG 15:53 05-10-2020



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\LP 2\dev\LP_2.exe

```
Enter the valid infix expression
a++b*c(/d)
Invalid
--------------------------------
Process exited after 36.19 seconds with return value 0
Press any key to continue . . .
```



C:\Users\sohan\Desktop\C Programs\Data Structures Lab\LP 2\dev\LP_2.exe

```
Enter the valid infix expression
((a*b)^(c/d))-g
the postfix exp is
ab*cd/^g-
```