Lab Program 2:-

2) 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;
 }}
```

T.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

```
#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;
```

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 15:52 05-10-2020

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

```
35  void infix_postfix(char infix[],char postfix[])
36  {
37  int top,i,j;
38  char s[30],symbol;
39  top=-1;
40  s[++top]='#';
41  j=0;
42  for(i=0;i<strlen(infix);i++)
43  {
44  symbol=infix[i];
45  while(F(s[top])>G(symbol))
46  {
47  postfix[j]=s[top--];
48  j++;
49  }
50  if(F(s[top])!=G(symbol))
51  s[++top]=symbol;
52  else
53  top--;
54  }
55  while(s[top]!='#')
56  {
57  postfix[j++]=s[top--];
58  }
59  postfix[j]='\0';
60  }
61  int main()
62  {int t;
63  char infix[20];
64  char postfix[20];
65  system("cls");
66  printf("Enter the valid infix expression\n");
67  scanf("%s",infix);
68  for(t=0;t<strlen(infix);t++)
69  {
70  if(infix[t]=='+'||infix[t]=='-'||infix[t]=='*'||infix[t]=='/'||infix[t]=='^'||infix[t]=='(')
71  {
72  if(infix[t+1]=='+'||infix[t+1]=='-'||infix[t+1]=='*'||infix[t+1]=='/'||infix[t+1]=='^'||infix[t+1]==')')
```

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.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

```
47  postfix[j]=s[top--];
48  j++;
49  }
50  if(F(s[top])!=G(symbol))
51  s[++top]=symbol;
52  else
53  top--;
54  }
55  while(s[top]!='#')
56  {
57  postfix[j++]=s[top--];
58  }
59  postfix[j]='\0';
60  }
61  int main()
62  {int t;
63  char infix[20];
64  char postfix[20];
65  system("cls");
66  printf("Enter the valid infix expression\n");
67  scanf("%s",infix);
68  for(t=0;t<strlen(infix);t++)
69  {
70  if(infix[t]=='+'||infix[t]=='-'||infix[t]=='*'||infix[t]=='/'||infix[t]=='^'||infix[t]=='(')
71  {
72  if(infix[t+1]=='+'||infix[t+1]=='-'||infix[t+1]=='*'||infix[t+1]=='/'||infix[t+1]=='^'||infix[t+1]==')')
73  { printf("Invalid");exit(0);
74  }
75  }
76  }
77  infix_postfix(infix,postfix);
78  printf("the postfix exp is\n");
79  printf("%s \n",postfix);
80  getch();
81  return 0;
82  }
83
```

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-
```