```
C-Program to convert Infix exPression to Postfix
 ex Pression: -
# include < stdio. b>
# include Lstring. h>
 gint F (char symbol)
    switch (symbol)
    (age '-): return 2; double
     case 1/2; return 4; # = [90+++
     cage (,'
    cage '$' : return 5 ; vi) aglides
     case ( ?; returno;
     (956, 1 : Lefan - 1: (200) 3 ) 3 (1)
     default: return 8;
                   POST FIXT & ] = Stack[ top
                     Stack[7-4609] - Bymb
      case 1, return 3
```

```
case '&'; return 6;
  case ( ; return 9;
   case ") ; returno;
   default: return 7;
void infix_Postfix (char infix[], char Postfix[])
  ist toP, i, d;
 char stack[30], symbol;
   toP=-1:
   Stack[++toP]= '#'
   1=0',
   for (i=o; iLstrlen (infix); i++)
                                 1 14 900)
    symbol=infix[i7:
    While (F(stack[toP])>G7(SymboD)
     Postfix[i] = Stack[top--];
    if (Flotack[toP]) != Gr (Symbol)
    Stack[++toP] = Symbol;
    else
    top__
```

```
while (stack[top]!= "#")
 Postfix[j++] = Stack [top-=];
Post fix[3]= 10);
char infix[20];
char Postfix [20];
Priviter the infix expression from user: (n").
 scanf ("%s", infix);
 : (xi)teof, xi)xi)teof _ xi)tui
  Printf ("the Bolfix expression is: (");
  Printf (" %s In", Postfix),
 return o;
```

```
#include<stdio.h>
 one.c
                       #include<string.h>
 prime.c
                       int F(char symbol)
 □ q8.c
                      □{
                  4
 □ q10.c
                  5
                             switch (symbol)
                  6
 n smalest.c
                  7
                                case '+':
'-':return 2;
                                case
 аттау.java
                  9
                                case
                                      '/':return 4;
  Roots.java
                 10
                                case
                 11
                                case
12
                                case 'S':return 5;
  Da.c
                 13
                                case '(':return 0;
  array.c
                                case '#': return -1;
                 14
                                default : return 8;
                 15
  Db.c
                 16
  D bin.c
                 17
  In cal.c.
                 18
                       int G(char symbol)
                 19
                      ₽{
  calculator.c
                            switch (symbol)
                 20
  D cie.c
                 21
                      Count.c
                 22
                            case '+':
  dd.c
                 23
                            case '-':return 1;
                                  1 % 1 .
                 24
                            case
  ex.java
                 25
                                 '/':return 3;
                            case
  exa.java
                                  1 ^ 1 .
                 26
                            case
  hash.c
                  27
                                 '&':return 6;
                            case
                  28
                                 '(':return 9;
                            case
   · indext.html
                  29
                            case ') ':return 0;
  matpro.c
                  30
                            default:return 7;
   new.c
                  31
                  32
   00.C
                       void infix postfix(char infix[], char postfix[])
                  33
   pass by.c
                  34
                      日{
    prec.c
                  35
                                int top, j, i;
                                char Stack[30], symbol;
                  36
   pri.c
                                top=-1;
    prime.c
                                Stack[++ton]='#':
   rand func.c ~
        gcc -Wall -c "prec.c" (in directory: C:\Users\DELL\Desktop\New folder)
        Compilation finished successfully.
Messages
```

```
prime.c
                             char Stack[30], symbol;
□ q8.c
                             top=-1;
                             Stack[++top]='#';
q10.c
                             j=0;
              39
smalest.c
                             for(i=0;i<strlen(infix);i++)
              40
~\De...JAVA)
              41
array.java
                                symbol=infix[i];
               42
                                while (F(Stack[top])>G(symbol))
               43
Roots.java
               44
~\Des...folder
                                  postfix[j]=Stack[top--];
               45
Da.c
                                  j++;
               46
array.c
               47
                                if(F(Stack[top])!=G(symbol))
               48
Db.c
                                Stack[++top]=symbol;
               49
bin.c
               50
                                else
Cal.c
               51
                                top--;
               52
acalculator.c
                              while (Stack[top] !='#')
               53
D de.c
               54
count.c
                                   postfix[j++]=Stack[top--];
                55
dd.c
                56
                              postfix[j]='\0';
                57
ex.java
                58
 exa.java
                59
                      int main()
 hash.c
                60
                    B{
                          char infix[20];
                61
 indext.html
                          char postfix[20];
                62
 matpro.c
                          printf ("Enter the infix expression from user: \n");
                63
 new.c
                          scanf("%s", infix);
                64
                          infix postfix (infix, postfix);
                65
 00.C
                          printf("The postfix expression is : \n");
                66
 pass by.c
                67
                          printf("%s\n",postfix);
   prec.c
                68
                          return 0;
                69
  pri.c
  prime.c
  rand func.c v
       gcc -Wall -c "prec.c" (in directory: C:\Users\DELL\Desktop\New folder)
       Compilation finished successfully.
```

int top, j, i;

one.c

Scribble

= {

34

C:\WINDOWS\SYSTEM32\cmd.exe

Enter the infix expression from user:

a^b*c-d+e/f/(g+h)

The postfix expression is :

ab^c*d-ef/gh+/+

(program exited with code: 0)

Press any key to continue . . .