

```

261 //Infix to postfix
262 #include<stdio.h>
263 #include<string.h>
264 int F(char symbol)
265 {
266     switch(symbol)
267     {
268         case '+':
269         case '-':return 2;
270         case '*':
271         case '/':return 4;
272         case '^':return 5;
273         case '$':return 0;
274         case '#':return -1;
275         default:return 8;
276     }
277 }
278 int G(char symbol)
279 {
280     switch(symbol)
281     {
282     case '+':
283     case '-':return 1;
284     case '*':
285     case '/':return 3;
286     case '^':
287     case '$':return 6;
288     case '(':return 9;
289     case ')':return 0;
290     default :return 7;
291     }
292 }
293 void infix_postfix(char infix[],char postfix[])

```

```

294 {
295     int top, i, j;
296     char s[30], symbol;
297     top=-1;
298     s[++top]='#';
299     j=0;
300     for(i=0;i<strlen(infix);i++)
301     {
302         symbol=infix[i];
303         while (F(s[top])>G(symbol))
304         {
305             postfix[j]=s[top--];
306             j++;
307         }
308         if (F(s[top])!=G(symbol))
309             s[++top]=symbol;
310         else
311             top--;
312     }
313     while (s[top]!='#')
314     {
315         postfix[j++]=s[top--];
316     }
317     postfix[j]='\0';
318 }
319 int main()
320 {
321     char infix[20];
322     char postfix[20];
323     printf("Enter a valid infix expression\n");
324     scanf("%s",infix);

```

```
325     infix_postfix(infix, postfix);  
326     printf("The postfix expression is: %s\n", postfix);  
327     printf("%s\n", postfix);  
328     return 0;  
329 }
```

Enter a valid infix expression

$A+B^C*D/F-G$

The postfix expression :

$ABC^D*F/+G-$

Process returned 0 (0x0)      execution time : 16.343 s

Press any key to continue.