

2] write a program to convert infix to postfix expression.

01-01-2024

2

```
#include <stdio.h>
```

```
#include <string.h>
```

```
int ind=0, pos=0, top=-1, length;
```

```
char temp, infix[20], postfix[20], stack[20]; Symbol;
```

```
void infixtopostfix();
```

```
void push(char symbol); ✓
```

```
char pop(char symbol); ✓
```

```
int pred(char symbol);
```

```
void main() {
```

```
printf("Enter the infix expression:");
```

```
scanf("%s", infix);
```

```
infixtopostfix();
```

```
printf("The infix expression is: %s\n", infix);
```

```
printf("The postfix expression is: %s\n", postfix); }
```

```
void push(char symbol) {
```

```
top = top+1;
```

```
stack[top] = symbol; }
```

```
char pop() {
```

```
char symbol;
```

```
symbol = stack[top];
```

```
top = top-1;
```

```
return(symbol); }
```

```
int pred(char symbol)
```

```
{ int p;
```

```
switch(symbol)
```

```
{ case 'x':
```

```
case '\': p=2;
```

```
break;
```

```
case '+':
```

```
case '-': p=1;
```

```
break;
```

```
case '(': p=0; break;
```

```
case '#': p=-1;
```

```
break;
```

```
} return(p); }
```

intopred (char symbol)

void infixtopostfix() {

length = strlen(infix);

push('#');

while(ind < length)

{ symbol = infix[ind];

switch (symbol)

{ case 'c': push(symbol);

break;

case ')': temp = pop();

while (temp != 'c')

{ postfix[pos] = temp;

pos++;

temp = pop(); }

break;

~~pop('c')~~

case '+':

case '-':

case '*':

case '/':

while (pred(Stack[top]) >= pred(symbol))

{ temp = pop();

postfix[pos] = temp;

pos++ ; }

push(symbol);

break;

default: postfix[pos++] = symbol; }

index++;

}

while (top > 0)

{ temp = pop();

postfix[pos++] = temp;

OUTPUT:

Enter the infix expression: $A * B + C * D - E$.

Infix expression: $A * B + C * D - E$.

Postfix expression: $AB * CD * + E -$.

End
11/1/2024

C:\Users\tanma\OneDrive\Di × + ▾

Enter the infix expression:A*B+C*D-E

Infix expression:A*B+C*D-E

Postfix expression:AB*CD*+E-

Process returned 29 (0x1D) execution time : 11.761 s

Press any key to continue.

|