

## 후위표기식 변환

코드

```
Infix_to_Postfix

126
127 void convert(char infix[], char postfix[])
128 {
129     stackType S;
130     init(&S);
131     char c, t;
132     int n = strlen(infix);
133     int z = 0;
134
135     for (int i = 0; i < n; i++)
136     {
137         c = infix[i];
138         switch (c)
139         {
140             case '+':
141             case '-':
142             case '/':
143             case '*':
144                 while (!isEmpty(&S) && prec(c) <= prec(peek(&S)))
145                 {
146                     postfix[z] = pop(&S);
147                     z++;
148                 }
149                 push(&S, c);
150                 break;
151             case '(':
152                 push(&S, c);
153                 break;
154             case ')':
155                 t = pop(&S);
156                 while (t != '(')
157                 {
158                     postfix[z] = t;
159                     z++;
160                     t = pop(&S);
161                 }
162                 break;
163             default:
164                 push(&S, c);
165                 break;
166         }
167     }
168     while (!isEmpty(&S))
169     {
170         postfix[z] = pop(&S);
171         z++;
172     }
173     postfix[z] = 0;
174     printf("\n");
175 }
176
```

실행결과

```
Microsoft Visual Studio 디버그 콘솔
(2+3)*4+9
29
C:\₩₩MU₩2-1₩₩자료구조₩₩실습₩₩과제₩₩스택₩₩Infix_to_P
가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

ex)  $(2+3)*4+9 = \text{Infix}$

convert(infix, postfix)  $\leftarrow$  infix (중위표기식)을 postfix(후위표기식)로 변환

$\Rightarrow s: \square$  스택 s를 초기화  $\Rightarrow \boxed{2} \quad 3 \Rightarrow \square \quad 23+ \Rightarrow \boxed{*} \quad 23+4$   
 $\Rightarrow \boxed{+} \quad 23+4* \Rightarrow \boxed{+} \quad 23+4*9 \Rightarrow \square \quad 23+4*9+9$   
ASCENDING  
문자열의 끝을  
나타냄

$\rightarrow \text{print}(\text{evaluate}(\text{postfix})) = 29$

# 회문

코드

```
회문 (전역)
74 //회문 판단
75 void palindrome(char k[])
76 {
77     StackType A;
78     init(&A);
79     StackType B;
80     init(&B);
81     int n = strlen(k);
82     for (int i = 0; i < n; i++)//문자열을 판독해서 전부 소문자로 만들어서 스택에 넣을
83     {
84         if (k[i] != ' '){//공백 제외
85             {
86                 if ('A' <= k[i] && k[i] <= 'Z')//대문자면 소문자로 만들고 push
87                 {
88                     char c = k[i] + 32;
89                     push(&A, c);
90                 }
91                 else if ('a' <= k[i] && k[i] <= 'z')//소문자면 push
92                 {
93                     char c = k[i];
94                     push(&A, c);
95                 }
96             }
97         }
98     }
99     int leng = Stack_length(&A)+1;//총 문자의 갯수
100     int mid = leng / 2;
101     for (int i = 0; i < mid; i++)
102     {
103         push(&B, pop(&A)); //B스택에 뒤 절반을 채워넣을
104     }
105     if (leng % 2 == 1)
106         char trash = pop(&A); //길이가 홀수면 중간값 버림
107     for (int i = 0; i < mid; i++)//A,B를 비교해서 서로 다른값이면 not palindrome 출력
108     {
109         char a = pop(&A);
110         char b = pop(&B);
111         if (a != b)
112         {
113             printf("not Palindrome");
114             break;
115         }
116         if (i == mid - 1)//끝까지 도착하면 회문
117             printf("palindrome");
118     }
119     //절반을 스택 B에 넣고 문자의 갯수가 홀수라면 중간값 버리고 A,B스택을 pop 하면서 비교
120 }
121
122
123
124 int main()
125 {
126     char Str[N];
127     gets(Str);
128     palindrome(Str);
129     return 0;
130 }
```

실행결과

```
eye
palindrome
C:\WKMJW2-1\자료구조\실습\과제\스
디버깅이 중지될 때 콘솔을 자동으
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르
```

```
race car
palindrome
C:\WKMJW2-1\자료구조\실습\과제\스
디버깅이 중지될 때 콘솔을 자동으
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르
```

ex) race car = str

palindrome(str)

A =  , b =   스택 A와 B를 초기화

for (0 → len(str)) 문자열의 길이만큼 반복

str[i]에 대해서 str[i]가 공백이 아닌 문자일 때

대문자는 + 32 (ASCII) 를 해서 대문자를 만들어 A에 push하고  
소문자는 그대로 A에 push한다.

A = r  
a  
c  
e  
c  
a  
r

그이후 for (0 → stack.length(A)) 스택의 절반만큼 반복

push(B, pop(A))로 A의 값 절반을 B에 채운다

A = e  
c  
a  
r    b = c  
a  
r ⇒ 그 후 총 문자의 개수가 홀수면 pop(A)로 맨 위 값을 버린다.

→ A = c  
a  
r    b = c  
a  
r ⇒ A와 B를 다 비울 때까지 pop(A)와 pop(B)를 비교해서 모두 같으면 회문. 값이 다르면 회문이 아님