

<문제>

중위 표기식을 입력 받아 후위 표기식으로 출력하시오.

EX) 3*4-2/1

<소스>

연산자 우선순위 부여 함수)

```
class PrefixOperatorApp
{
    참조 2개
    public static int Prec(char c)
    {
        switch (c)
        {
            case '+':
            case '-':
                return 1;

            case '*':
            case '/':
                return 2;

            case '(':
                // 실제로는 우선순위가 더 높지만 왼쪽 괄호가 top인 경우 상관없이 push해야하므로
                return 0;
        }

        return -1;
    }
}
```

메인)

```
public static void Main()
{
    StringBuilder input = new StringBuilder();    //문자열을 이어 붙이기 위해 StringBuilder 사용
    StringBuilder answer = new StringBuilder();
    Stack<char> st = new Stack<char>();

    Console.WriteLine("중위 표기식으로 식을 작성하시오 : ");
    string tmp = Console.ReadLine();
    input.Append(tmp);
}
```

메인 중 입력에 따른 반복문)

```

for (int i = 0; i < input.Length; i++)
{
    char ch = input[i];
    switch (ch)
    {
        // 연산자를 만나면 스택 상단의 연산자와 우선순위 비교
        // 지금 넣으려는 연산자의 우선 순위가 더 크면 해당 연산자를 스택에 삽입
        // 작거나 같으면 스택 상단의 연산자를 변환 문자열에 추가하고 다시 반복
        case '+':
        case '-':
        case '*':
        case '/':
            while (st.Count() != 0 && Prec(ch) <= Prec((char)st.Peek()))
            {
                answer.Append(st.Peek().ToString());
                st.Pop();
            }
            st.Push(ch);
            break;

        // 왼쪽 괄호가 나오면 스택에 삽입
        case '(':
            st.Push(ch);
            break;

        // 오른쪽 괄호가 나오면 스택에서 왼쪽 괄호가 나올때까지
        // 모든 연산자를 변환 문자열에 추가하고 pop. 왼쪽 괄호는 삭제
        case ')':
            while (st.Count() != 0 && (char)st.Peek() != '(')
            {
                answer.Append(st.Peek().ToString());
                st.Pop();
            }
            st.Pop(); //왼쪽 괄호
            break;
    }
}

```

```

// 피연산자는 바로 변환 문자열에 추가
default:
    answer.Append(ch.ToString());
    break;
}
}

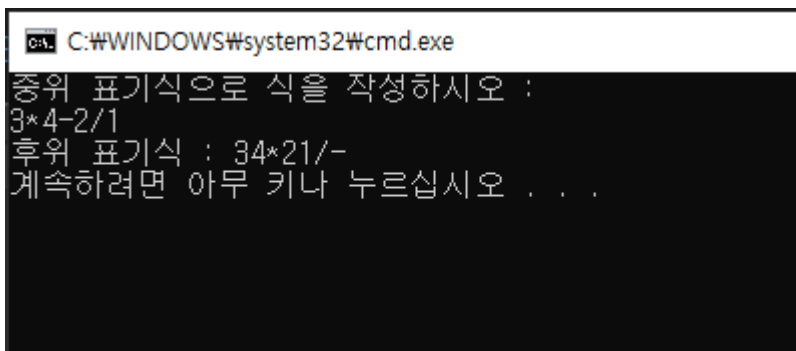
```

스택에 남아있는 모든 연산자 POP)

```
//처리할 문자가 표기식에 남아있지 않다면 스택에 남아있는 모든 연산자 변환 표기식에 추가와 pop
while (st.Count != 0)
{
    answer.Append(st.Peek().ToString());
    st.Pop();
}

Console.WriteLine("후위 표기식 : " + answer);
}
```

<결과>



```
C:\WINDOWS\system32\cmd.exe
중위 표기식으로 식을 작성하시오 :
3*4-2/1
후위 표기식 : 34*21/-
계속하려면 아무 키나 누르십시오 . . .
```

<설명>

중위 표기식을 후위 표기식으로 표현하기 위해 연산자에 우선순위를 적용하는 함수를 사용하였다. Prec라는 함수를 통해 곱셈 문자와 나눗셈 문자를 파라미터로 받으면 2를 리턴하고, 덧셈 문자와 뺄셈 문자를 받으면 1을 리턴한다.

메인 함수 중, 입력 받은 문자의 경우에 따라 switch문을 통해 분기되는데 사칙연산 문자를 받으면 스택 상단의 연산자와 비교하여, 지금 넣으려는 연산자의 우선순위가 더 크면 해당 연산자를 스택에 삽입하고, 작거나 같으면 스택 상단의 연산자를 출력하고 다시 반복한다. 좀 더 다양한 경우를 대비하여 괄호 기호를 입력 받았을 상황을 대비하여 왼쪽 괄호가 나오면 스택에 삽입하고, 오른쪽 괄호가 나오면 스택에서 왼쪽 괄호가 나올 때까지 모든 연산자를 POP한다. 그 후, 왼쪽 괄호는 삭제를 한다.

처리할 문자가 표기식에 남아있지 않다면 스택에 남아있는 모든 연산자를 후위 표기식에 추가 한 후 POP처리를 하여 스택을 비워준다. 자세한 변환 과정은 아래 그림을 통해 설명하겠다.

중위 표기 수식	연산자 스택	후위 표기 수식
3 * 4 - 2 / 1		3
3 * 4 - 2 / 1	*	3
3 * 4 - 2 / 1	*	34
3 * 4 - 2 / 1	* -	34
	-	34*
3 * 4 - 2 / 1	-	34*2
3 * 4 - 2 / 1	- /	34*2
3 * 4 - 2 / 1	- /	34*21
	-	34*21/
		34*21/-

3 4 * 2 1 / -

↑ 피연산자는 스택에 삽입

3

3 4 * 2 1 / -

4
3

3 4 * 2 1 / -

↑ 3 * 4 연산자 결과를 삽입

--

12

$$3\ 4\ * \ 2\ 1\ /\ -$$

2
12

$$3\ 4\ * \ 2\ 1\ /\ -$$

1
2
12

$$3\ 4\ * \ 2\ 1\ /\ -$$

2
12

$$3\ 4\ * \ 2\ 1\ /\ -$$

10

↑ 최종 연산 결과