

REPORT

**자료구조**

A반 202144023 최지혁 | 컴퓨터 정보과 | 2022.04.14

# -목 차-

1. 문제설명
2. 요구사항 분석
3. 요구사항 명세서
4. 설계

1)알고리즘 설계

2)입출력 설계

1. Program 작성
2. 출력 Result
3. 참고문헌
4. 검토

**PART I 문제 설명**

I-I) 문제

중위(infix)표기법으로 작성된 수식을 입력 받아 주어진 처리 조건에 맞게 이를 후위(postfix)표기법으로 변환하는 과정과, 후위(postfix)표기법 변환된 수식을 계산하여 결과를 출력하는 과정에 대한 알고리즘을 제시하고, C언어 또는 Java언어로 실행 과정을 구현하는 문제이다.

I-II) 문제 분석

1. 변환과정 (스택)

스택은 한쪽 끝에서만 자료를 넣거나 뺄 수 있는 선형 구조로 되어 있습니다. 식당에 쌓여있는 접시들이 좋은 예입니다. 순서대로 쌓인 접시가 스택 구조와 같습니다. 접시가 필요하면 제일 위에 있는 접시부터 사용하며 가장 아래 있는 접시는 마지막에 사용됩니다.

* **특징**

스택은 LIFO (Last In First Out) / FILO (First In Last Out) 순서를 따릅니다.

* LIFO : 마지막으로 들어온 값이 처음으로 나가는 것
* FILO : 처음 들어온 값이 마지막에 나가는 것

스택은 완전히 꽉 찼을 때 Overflow 상태라고 하며 완전히 비어 있으면 Underflow 상태라고 합니다.

삽입(Push)과 제거(Pop)는 모두 Top이라는 스택의 한쪽 끝에서만 일어납니다.

1. 프로그래밍 언어 (C언어)

C언어는 현재 사용하고 있는 거의 모든 컴퓨터 시스템에서 사용할 수 있는 프로그래밍 언어입니다.

**- 특징**

**장점**

C언어로 작성된 프로그램은 다양한 하드웨어로의 이식성이 좋습니다.

C언어는 절차 지향 프로그래밍 언어로, 코드가 복잡하지 않아 상대적으로 유지보수가 쉽습니다.

C언어는 저급 언어의 특징을 가지고 있으므로, 어셈블리어 수준으로 하드웨어를 제어할 수 있습니다.

C언어는 코드가 간결하여, 완성된 프로그램의 크기가 작고 실행 속도가 빠릅니다.

**단점**

C언어는 저급 언어의 특징을 가지고 있으므로, 자바와 같은 다른 고급 언어보다 배우기가 쉽지 않습니다.

C언어는 다른 언어와는 달리 시스템 자원을 직접 제어할 수 있으므로, 프로그래밍하는데 세심한 주의를 기울여야 합니다.

**PART II**  요구사항분석

1. 처리 조건

① 사용 연산자는 “덧셈(+), 뺄셈(-), 곱셈(\*), 나눗셈(/), 나머지(%)” 로 한정한다.

② 피연산자는 “0-9” 사이의 한자리수의 정수만을 사용 한다.

③ 수식에 괄호 “ (...)” 를 사용한다.

④ infix를 postfix로 변환하는 과정을 출력하고 결과를 제시한다.

⑤ postfix표기법으로 제시된 수식표기법을 계산하는 과정을 출력하고결과를 제시한다.

⑥ 위 <처리 조건> ①~③에서 나올 수 있는 경우의 수를 모두 제시 한다.

1. 처리 조건 분석

1)- ① 사용 연산자는 “덧셈(+), 뺄셈(-), 곱셈(\*), 나눗셈(/), 나머지(%)” 로 한정

**-연산자의 우선순위**

1. (…)가 있을 시에는 (…) 먼저 계산한다.
2. 연산자 ‘\*’의 우선 순위 첫번째로 한다
3. 연산자 ‘/’, ’%’, 는 우선순위를 ‘\*’과 동일하게 첫번째로 한다
4. 연산자 ‘+’의 우선 순위를 두번째로한다
5. 연산자 ‘-‘의 우선 순위를 ‘+’와 동일하게 두번째로 한다

**1)- ② 피연산자는 “0-9” 사이의 한자리수의 정수만을 사용 한다.**

* 피연산자 정수

1. 피연사자가 실수가 들어왔을 경우에는 오류 처리해준다
2. 피연산자가 0보다 작은 수가 들어올 경우에는 오류 처리해준다
3. 피연산자가 9보다 큰 수가 들어올 경우에는 오류 처리해준다

**1)- ③ 수식에 괄호 “ (...)” 를 사용한다.**

**1)- ④ infix를 postfix로 변환하는 과정을 출력하고 결과를 제시한다.**

**후위 표기법은 컴퓨터가 연산을 하기 쉽게 표현하는 방법이다.**

* **중위표기법 -> 후위표기법**

중위 표기법: 연산자를 피연산자의 가운데에 표기하는 방법

후위 표기법: 연산자를 피연산자 뒤에 표기하는 방법

<피연산자는 출력하고, 연산자는 stack에 push 한다. 이후 오른쪽 괄호가 나오면 POP)

**중위 표기법을 후위 표기법으로 바꾸는 알고리즘**

1. 중위 표기식에서의 연산자, 피연산자 간 우선순위를 괄호를 묶는다
2. 왼쪽 괄호를 만나면 무시하고 다음 문자를 읽는디
3. 연산자를 만나면 스택에 push한다
4. 오른쪽 괄호를 만나면 스택을 pop하여 출력한다
5. 수식이 끝나면 스택이 공백이 될 때까지 pop하여 출력한다

예) infix -> (2+3)\*5

|  |  |  |
| --- | --- | --- |
| symbol | 연산자 stack | postfix 표기법 |
| ( | ( |  |
| 2 | ( | 2 |
| + | (+ | 2 |
| 3 | (+ | 23 |
| ) |  | 23+ |
| \* | \* | 23+ |
| 5 | \* | 23+5 |
| \0 |  | 23+5\* |

postfix -> 23+5\*

**1)- ⑤ postfix표기법으로 제시된 수식표기법을 계산하는 과정을 출력하고 결과를 제시한다.**

* **후위표기법 -> 수식표기법**

1. 숫자는 스택에 그냥 추가한다.

2. 연산자가 나오면 숫자 2개를 pop 해서 계산한다.

3. 이때 먼저 pop 되는 숫자가 두 번째 값, 나중에 pop되는 숫자가 첫 번째 값으로 하여 계산해야 한다. 계산한 값은 다시 스택에 넣는다.

예) 23+5\*

|  |  |  |
| --- | --- | --- |
| symbol | 연 산 | 피연산자 stack |
| 2 |  | 2 |
| 3 |  | 23 |
| + | 2+3 | 5 |
| 5 |  | 55 |
| \* | 5\*5 | 25 |
| \0 |  |  |

계산결과 -> 25

1. **⑥ 위 <처리 조건> ①~③에서 나올 수 있는 경우의 수를 모두 제시 한다.**

괄호가 없는 경우 2+3\*5

괄호가 있는 경우 (2+3)\*5

복수의 연산자들이 우선순위가 같을 경우, 다른 경우 2+3+5, 2%3\*5 등

**PART III**  시스템 명세

1. 함수

메인 함수 main(): 함수의 메인으로 프로그램이 시작이 되는 함수이며, 이번 프로그램에서는 수를 받아오는 역할과 최종 값을 출력하는 역할을 한다.

기호 구분 함수 sign(): 메인함수에서 받아 온 값들중에 요구 사항조건에 수식을 한정 지어주었기 때문에 요구 사항에 맞는 수식만 사용하도록 검사하는 함수이다.

우선 순위지정 Priority(): 수식의 우선 순위를 지정을 해주지 않으면 오류가 나거나 잘못된 값이

출력되기 때문에 수식의 우선 순위를 지정해주어 프로그램의 오류 발생을 막기 위한 함수이다.

스택 원소 삽입 push() : 스택에 원소를 삽입해주기 위한 함수이다.

중위 표기법 -> 후위 표기법 infixTopostfix(): 중위 표기법을 후기 표기법으로 변경하여 컴퓨터의 연산 방법에 맞추기 위한 함수이다.

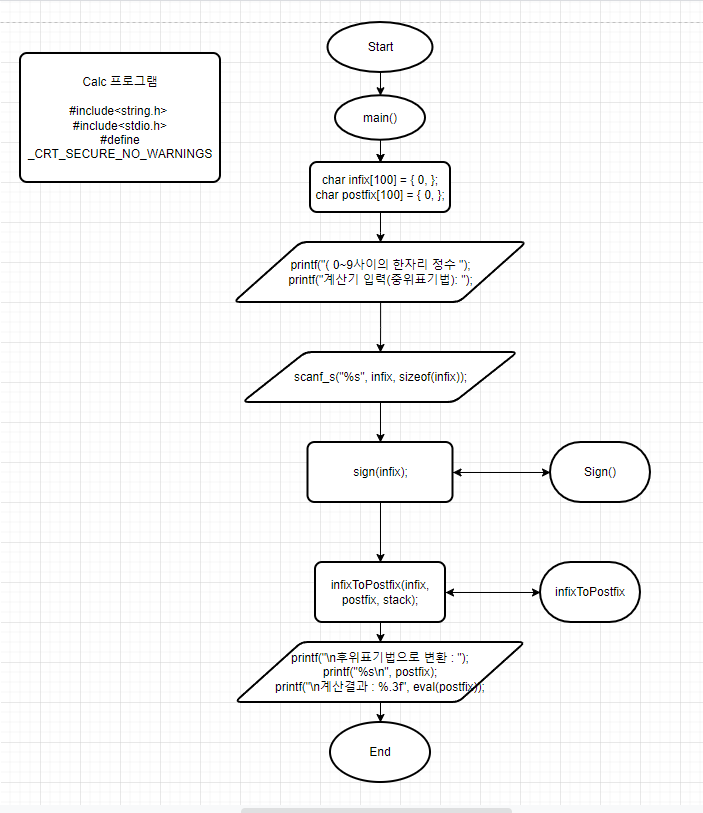
후위 표기법 -> 수식 표기법 eval(): 후위 표기법으로 변경된 값들을 계산을 통해 실제 값을

구하기 위한 함수이다.

**PART IV**  시스템 명세

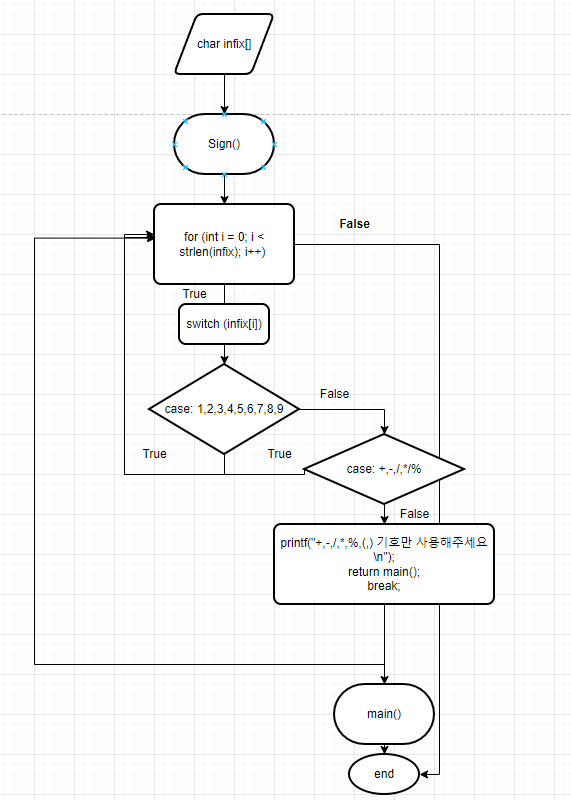
1. 알고리즘 설계

Main()함수 순서도



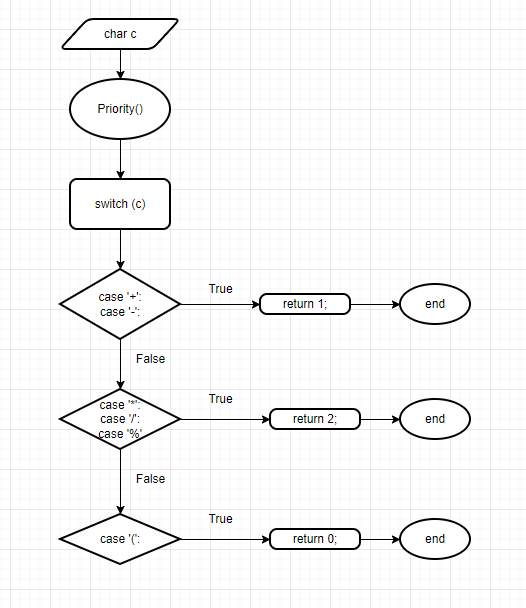
1. **Infix와 postfix 저장 공간을 만들어준다.**
2. **계산기의 입력값을 받는다.**
3. **받은 값을 sign()로 넘겨 수식이 올바른지 검사받는다**
4. **받은 값을 infixtoPostfix()함수에 넘겨 중위 표기법을 후기표기법으로 변경한다.**
5. **값들을 출력해준다.**

sign()함수 순서도



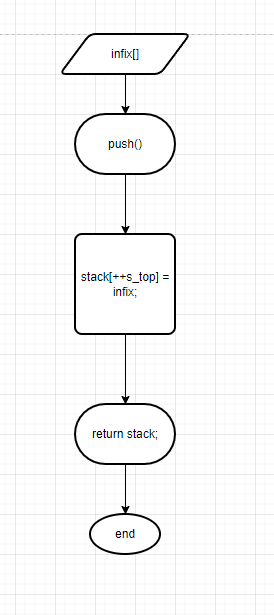
1. **Main()함수로부터 infix값을 받아온다**
2. **For문을 통해 switch문으로 들어간다**
3. **Switch-case문을 통해 수식을 검사 맡는다**
4. **이상이 없으면 종료한다**
5. **이상이 있으면 다시 main()문으로 리턴한다.**

Priority()함수 순서도



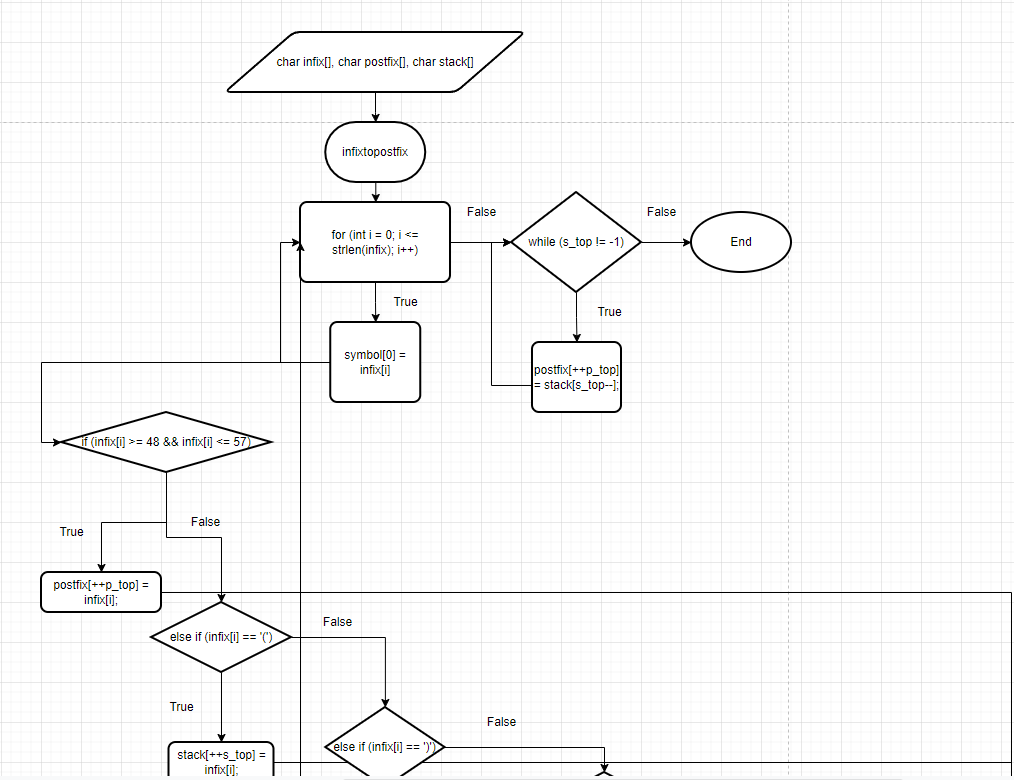
1. **infixtoPostfix()함수로부터 c값을 받아온다.**
2. **Switch – case문을 통해 각자의 리턴번호를 부여받아 우선 순위를 정한다**
3. **값들을 리턴해준다**

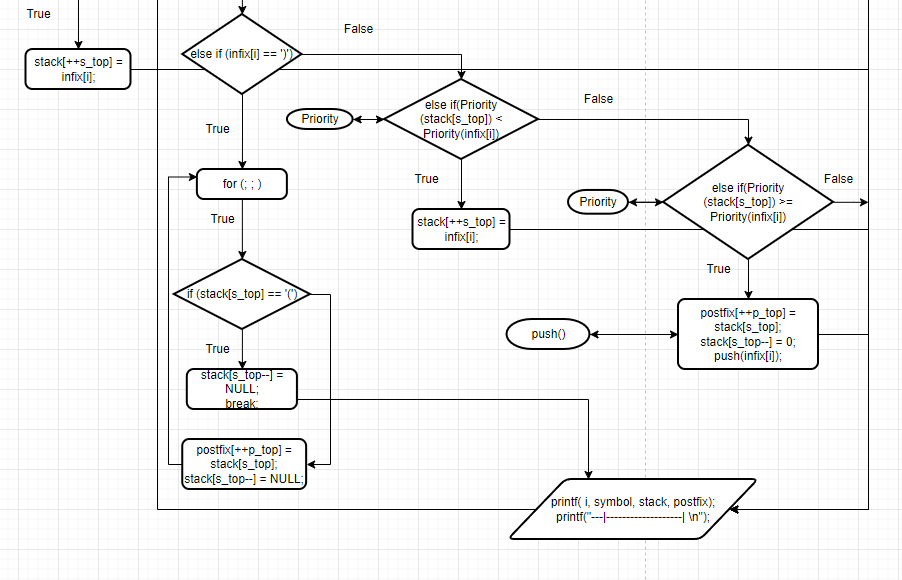
Push()함수 순서도



1. **infixtoPostfix()함수로부터 infix값을 받아온다.**
2. **만들어진 스택 영역에 값을 저장한다.**

Infixtopostfix()함수 순서도

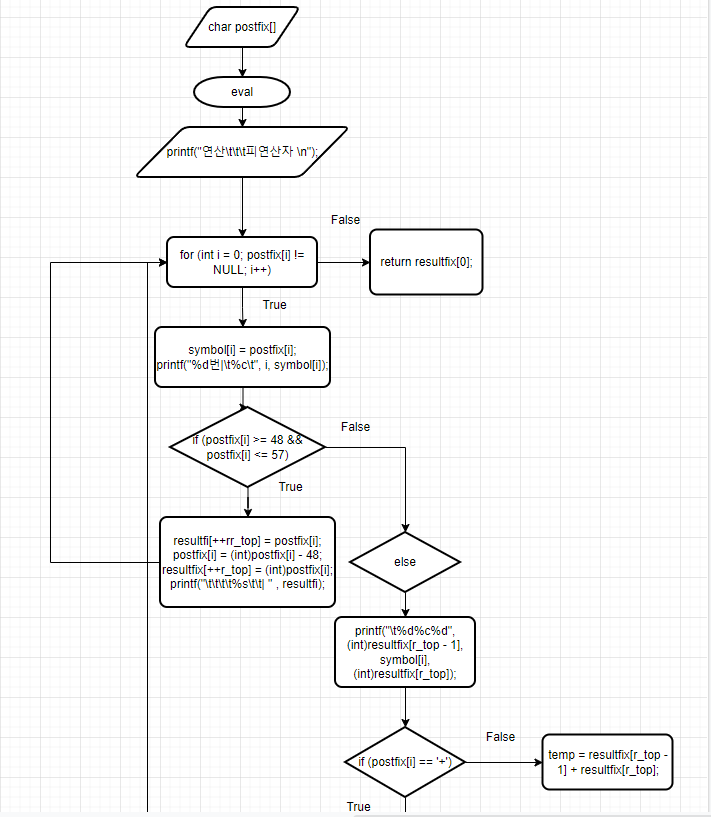


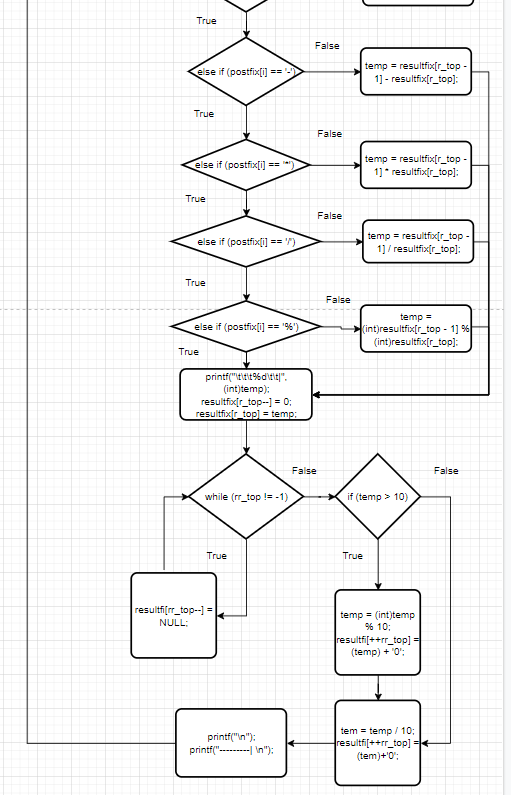


Infixtopostfix()함수 순서도-2

1. **main()함수로부터 infix[],postfix[].stack[]값을 받아온다.**
2. **For문을 통해 받아온 infix의값들을 검사한다**
3. **If문을 통해 받아온 값이 수인지 수식인지 검사한다**
4. **값이 수라면 postfix영역에 넣어준다**
5. **(가 들어온 경우에는 stack 영역에 넣어준다**
6. **수식이 들어온 경우도 stack 영역에 넣어준다.**
7. **)가 들어온 경우에는 stack의 값들을 postfix에 넣어준다**
8. **그 후에 stack영역을 초기화 시켜준다.**

eval()함수 순서도-1

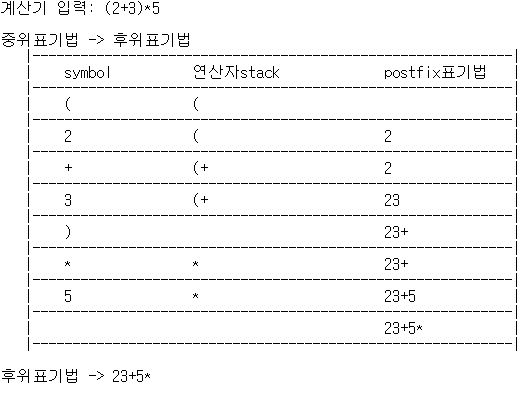




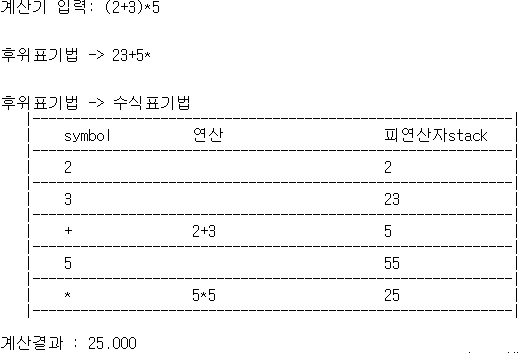
eval()함수 순서도 -2

1. **main()함수로부터postfix[]값을 받아온다**
2. **postfix의 값이 수식인지 수인지 if문을 통해 검사한다**
3. **postfix의 값이 수일 경우 현재 postfix의 값은 문자이기 때문에 문자형을 숫자형으로 변경해준다**
4. **변경된 값을 resultfix영역에 넣어준다**
5. **postfix의 값이 수식일 경우 resultfix영역을 이용하여 계산후에 값을 temp에 저장시킨다.**
6. **그리고 다시 temp의 값을 resultfix영역에 넣어준다**
7. **최종 계산이 끝나면 resultfix의 값을 리턴시켜준다.**
8. 출력 설계

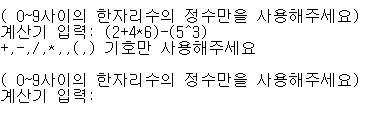
중위표기법 -> 후위표기법 변경



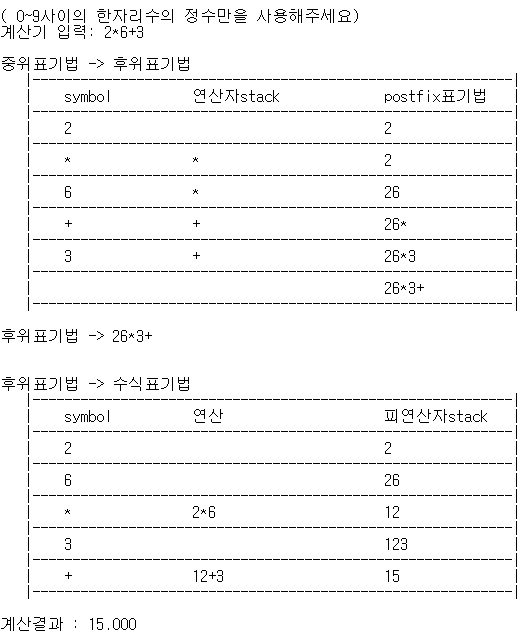
후위 표기법 -> 수식 표기법 변경



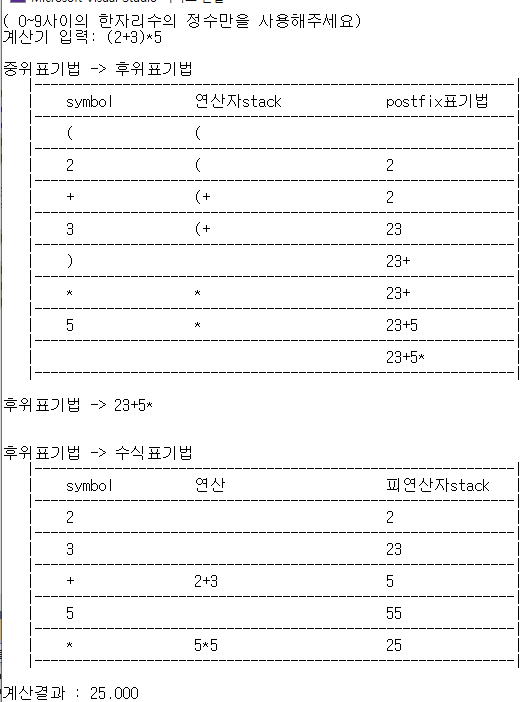
올바르지 않은 수식이 왔을 경우



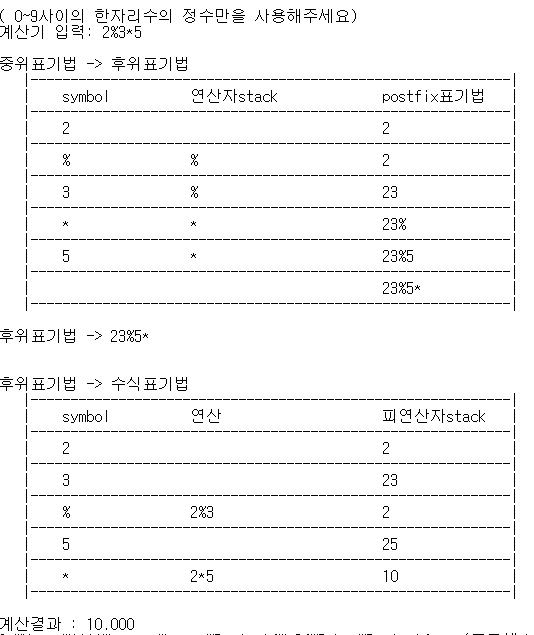
괄호가 없는 경우



괄호가 있는 경우

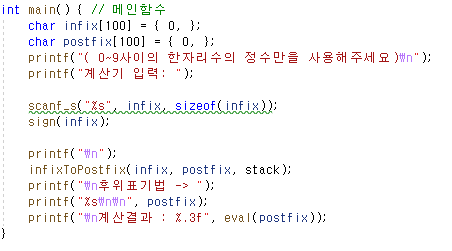


우선 순위가 같은 경우

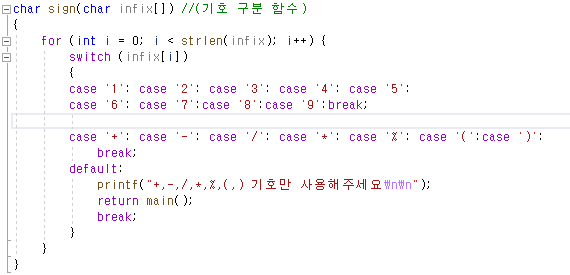


**PART IIV**  프로그램 작성

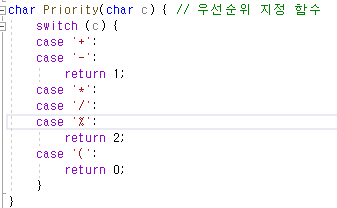
main()함수



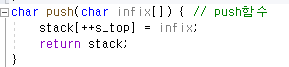
sign()함수

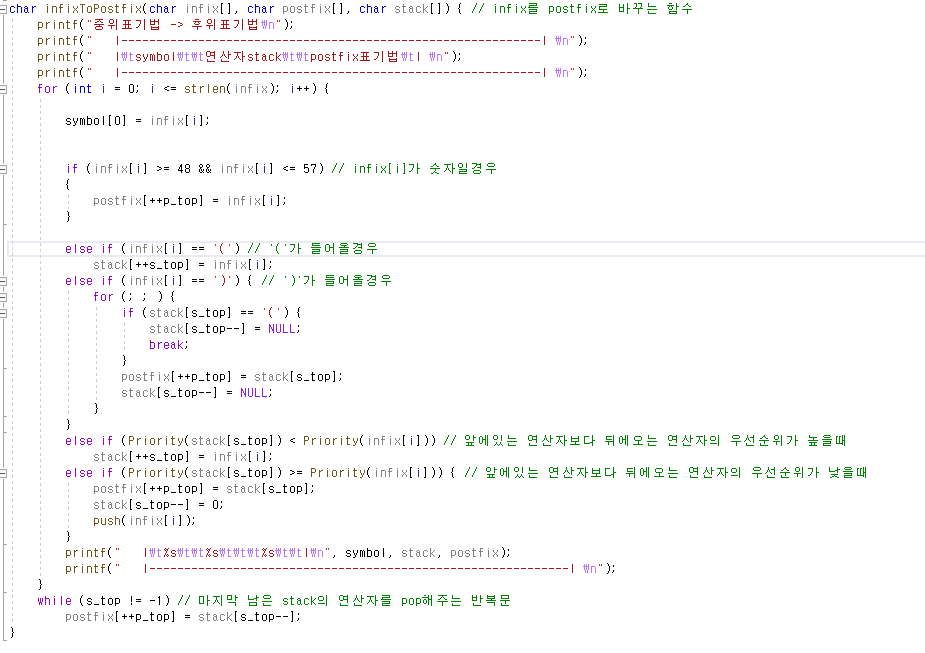


Priority()함수

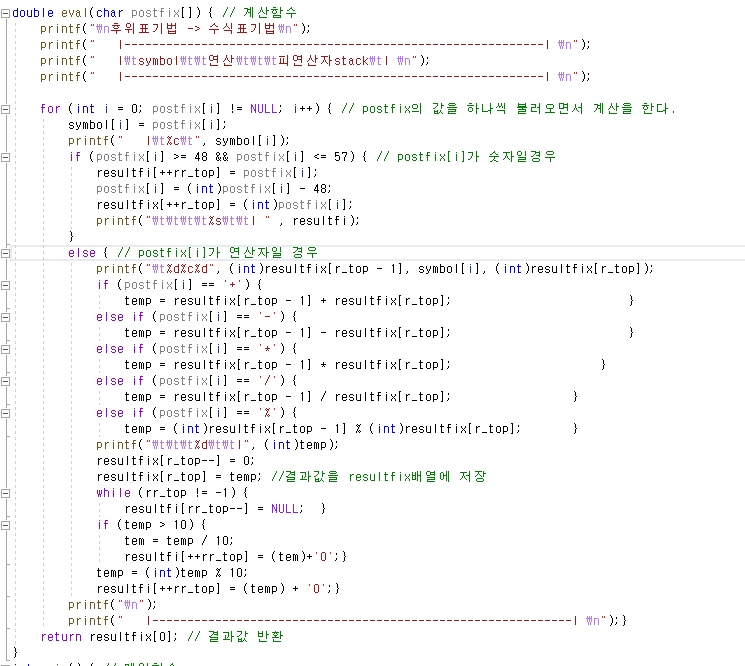


Push()함수

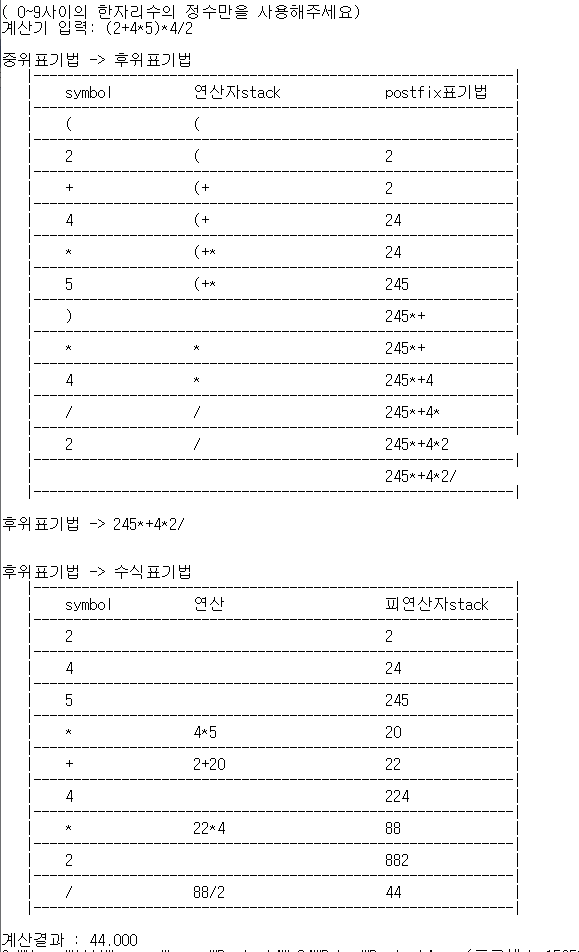


intopostfix()함수

eval()함수



**PART VI 출력 Result**

****

**PART VII 참고 문헌**

**중위표기식 -> 후위표기식:** [**https://dev-with-precious-dreams.tistory.com/entry**](https://dev-with-precious-dreams.tistory.com/entry)

**C언어란 :** [**http://www.tcpschool.com/c/c\_intro\_basic**](http://www.tcpschool.com/c/c_intro_basic)

**후기 표기식 -> 수식 표기식:** [**https://todaycode.tistory.com/73?category=997273**](https://todaycode.tistory.com/73?category=997273)

**정수->문자:** [**https://www.delftstack.com/ko/howto/c**](https://www.delftstack.com/ko/howto/c)

**PART VII 검토**

**(2+3)\*5- : 이런 값들을 예외 처리 못함**

**0~9사이의 정수의 값이 아닌 다른 값이 들어와도 예외 처리 못함**