

HW3

• 문제 1:

교재에 주어진 eval과 postfix 함수는 (Program 3.13, 3.15)

$+$, $-$, $/$, $*$, $%$ 등 5개의 operator에 한해서만 동작한다.

이를 수정하여 **unary operator인 $-$ 까지 포함**할 수 있도록 만들고자 한다.

(단, 구분을 위해서 **postfix에서는 $-$ 대신 $\#$ 로 표시한다고 가정**하자.)

unary operator $-$ 와 $+$, $-$, $/$, $*$, $%$ 를 포함하는 수식이 입력으로 들어왔을 때, 해당 수식을 postfix로 바꾼 결과와, postfix 식의 계산 결과를 출력하는 프로그램을 작성하시오.

- 문제 1:

예제	
입력	출력
Input: -6	Postfix: 6# Result: -6
Input: $(1 - (-3) - 5)$	Postfix: 13#-5- Result: -1
Input: $3 * 2 + 4 * (5 - 1)$	Postfix: 32*451-*+ Result: 22

- **문제 1:**

- **입력:**

- 최소 1글자, 최대 20글자 길이의 string이 입력으로 들어옴
 - Unary operator $-$ 를 포함하는 것 외에 기타 제약 사항 (예, operand가 single digit라고 가정 등)은 교재에 나와있는 것과 동일함

- **출력:**

- 첫 줄에 주어진 infix 식을 postfix로 바꾼 식
 - 두 번째 줄에 해당 postfix 식의 계산 결과

- 문제 1:

- 제약 조건:

- 모든 입출력은 stdin, stdout 사용
 - 전역변수 사용 불가
 - 정적할당 가능
 - push/pop 함수를 작성할 것
 - 계산 결과는 반드시 postfix 수식을 계산한 것이어야 함
(Infix 수식을 계산하여 출력한 경우 점수 없음)

• 문제 2:

수식을 표현하는 방식으로는 infix, postfix 뿐만 아니라 prefix도 존재한다.

Infix를 prefix로 바꾸는 간단한 방법은 아래와 같다:

Step1: Infix 수식을 반전한다.

e.g., $3+2*4 \rightarrow 4*2+3$

Step2: 반전된 수식에 대해, postfix 수식을 구한다.

e.g., $4*2+3 \rightarrow 42*3+$

Step3: 구해진 postfix 수식을 다시 반전한다.

e.g., $42*3+ \rightarrow +3*24$

그러나 이는 이미 알고 있는 내용들이기 때문에, 새로운 방식을 사용하여 infix를 prefix로 바꿔 보고자 한다. postfix 함수를 사용하지 않고, stack을 사용하여 infix 수식을 prefix 수식으로 바꾸어 주는 프로그램을 작성하시오.

- 문제 2:

예제	
입력	출력
Infix: $3 * 8 + 7 / 1$	Prefix: $+ * 3 8 / 7 1$
Infix: $(4 - 9 / 5) * (4 / 1 - 2)$	Prefix: $* - 4 / 9 5 - / 4 1 2$

- 문제 2:

- 입력:

- 최소 1글자, 최대 20글자 길이의 string이
입력으로 들어옴
 - 기타 제약 사항 (예, operand가 single digit라고
가정 등)은 교재에 나와있는 것과 동일함

- 출력:

- 주어진 infix 식을 prefix로 바꾼 식

- **문제 2:**

- 제약 조건:

- 모든 입출력은 stdin, stdout 사용

- 전역변수 사용 불가

- 정적할당 가능

- **필요한 stack의 ADT가 구현** 되어있어야 함. 예:

- push/pop함수

문제 3:

사용자는 숫자 num과 삭제할 자리 수 k를 입력한다.
num에서 자리 수를 k개 만큼 뺏을 때 가장 작은 숫자를 구하는 프로그램을
반드시 stack을 사용하여 작성하시오. 예를 들어 num이 13425고 k가 2일 때,
num에서 2개의 자리를 뺏을 때 가장 작은 숫자는 125이다.

문제 3:

예제	
입력	출력
1432219 3	1219
10200 1	200
10 2	0

문제 3:

입력:

- 첫 번째 줄 : num 입력
- 두 번째 줄 : 입력받은 num에서 뺄 자리의 개수 k

출력:

- num에서 k만큼의 자리 개수를 뺐을 때 가장 작은 수

문제 3:

제약 조건:

- 모든 입출력은 stdin, stdout 사용
- 전역변수 사용 불가
- **필요한 stack의 ADT 함수가 구현** 되어있어야 함.

제출 방법

과제2와 동일. 단지 과제 번호만 3으로 수정할 것.