

국어  
수능특강

적용학습 기술 1. 스택 알고리즘의 이해와 적용 <EBS변형문제>

이 콘텐츠는 「콘텐츠산업 진흥법」에 따라 최초 제작일부터 5년간 보호됩니다.  
본 콘텐츠의 무단 배포 시, 콘텐츠산업 진흥법에 의거하여 책임을 질 수 있습니다.

\*지문분석

(가) 컴퓨터에서는 데이터를 처리하기 위해 여러 가지 데이터 구조를 활용한다. 그 대표적인 것으로 ‘스택 (stack)’을 들 수 있다. 스택은 한쪽 끝에서만 데이터가 들어가거나 나올 수 있는 특성[스택의 특성]을 가지는데, 스택에서 데이터가 가장 먼저 나올 수 있는 위치를 톱(top)[**‘톱’의 개념**]이라 한다. 스택의 구조는 위쪽만 뚫려 있는 통 안에 책을 차곡차곡 쌓는다고 생각하면 쉽게 이해할 수 있다. 만약 n개의 책을 통에 쌓았다면 가장 먼저 꺼낼 수 있는 책은 아래에서부터 n번째에 있는 것이고, 그 위에 책을 또 올리게 되면 그 책은 아래에서부터 n+1번째에 있게 된다. 스택 구조는 이와 같은 통에, 톱은 가장 위에 있는 책의 위치에 비유될 수 있다. 그리고 스택 알고리즘은 데이터가 들어가 보관되고 나오는 규칙으로[스택 알고리즘의 개념], 통에 책을 넣어 쌓고 책을 꺼내는 행위에 비유될 수 있다.

스택과 스택 알고리즘의 특성

➔ 스택 구조 = 책을 넣는 통, 톱 = 통에서 가장 위에 있는 책의 위치, 스택 알고리즘 = 책을 넣어 쌓고 꺼내는 행위

(가) 단락의 독해 포인트 : 스택의 구조와 톱, 스택 알고리즘을 통에 책을 넣고 꺼내는 것에 빗대어 표현.

(나) 컴퓨터 내의 데이터 처리에서 스택 알고리즘은 수식 변환의 과정에 활용된다.[스택 알고리즘의 쓰임] 수식은 ‘+’(덧셈), ‘\*’(곱셈), ‘-’(뺄셈), ‘/’(나눗셈) 등과 같은 연산자와, ‘x, y, z, A, B, C’ 등과 같은 변수나 ‘1, 2, 3’ 등과 같은 상수가 포함되는 피연산자로 구성된다.[수식의 구성 요소] 그리고 수식은 크게 세 가지 표기 식으로 표현할 수 있는데, 연산자가 피연산자 가운데 있는 중위 표기 식, 연산자가 피연산자 앞에 있는 전위 표기 식, 연산자가 피연산자 뒤에 있는 후위 표기 식이 그것이다. 예를 들어 ‘x 더하기 y 나누기 z’는 중위 표기 식으로는 ‘x +y/z’로, 전위 표기 식으로는 ‘+x/yz’로, 후위 표기 식으로는 ‘xyz/ +’로 표현된다. 우리가 일상생활에서 사용하는 것은 중위 표기 식이지만, 컴퓨터 내부에서는 괄호를 사용하지 않고도 계산해야 할 순서를 알 수 있는 후위 표기 식을 활용하기 때문에 중위 표기 식으로 입력한 수식은 스택 알고리즘을 통해 후위 표기 식으로 변환되어 계산된다.

수식 변환의 과정에 활용되는 스택 알고리즘

- ➔ 수식의 표기 식 ① 중위 표기 식 ② 전위 표기 식 ③ 후위 표기 식
- ➔ 컴퓨터 내부에서는 후위 표기 식을 활용 => 우리가 일상생활에서 사용하는 중위 표기 식을 입력하면 스택 알고리즘을 통해 후위 표기 식으로 변환되어 계산.

(나) 단락의 독해 포인트 : 세 가지 수식의 표기 식을 이해하기 쉽게 예를 들어 설명.

(다) 중위 표기 식을 후위 표기 식으로 변환하는 방법은 간단하다. 기본적인 전제는 피연산자들의 순서는 변하지 않는다는 것이다. 즉 중위 표기 식을 연산자 기준으로 괄호로 묶은 다음, 각 연산자를 묶고 있는 괄호의 오른쪽 괄호 밖으로 연산자를 이동한 후 괄호를 모두 제거하면 된다.[중위 표기 식을 후위 표기 식으로 변환하는 방법] 예를 들어 중위 표기 식 ‘A+B\*C’를 후위 표기 식으로 바꾸기 위해서는 ‘(A+(B\*C))’로 괄호를 묶은 뒤, ‘(A(BC)\*)+’와 같이 괄호의 오른쪽에 연산자를 옮긴 후 ‘ABC\* +’와 같이 괄호를 제거하면 된다. 중위 표기 식을 전위 표기 식으로 변환하는 과정 역시 동일하다. 다만 연산자를 오른쪽에 옮기는 것이 아니라 왼쪽으로 옮기는 것만 차이가 있다.[예를 들어 ‘(A+(B\*C))’를 전위 표기 식으로 변환하면 ‘+A\*(BC)’ -> 괄호를 제거하면 ‘+A\*BC’]

중위 표기 식을 후위 표기 식으로 변환하는 방법

- ➔ 중위 표기 식을 후위 표기 식으로 변환 과정 A+B\*C
- ① 연산자 기준으로 괄호로 묶는다.(A+(B\*C))
- ② 연산자를 묶고 있는 괄호의 오른쪽 밖으로 연산자를 이동한다.(A(BC)\*)+

③ 괄호를 모두 제거한다.ABC\*\*

※이 때 피연산자들의 순서는 변하지 않도록 한다.

(다) 단락의 독해 포인트 : 중위 표기 식을 후위 표기 식으로 변환하는 과정을 예를 통해 설명.

(라) 컴퓨터에서는 중위 표기 식을 후위 표기 식으로 변환하기 위해 앞에서 언급한 스택 알고리즘을 활용한다. **후위 표기 식에서 피연산자들의 순서는 변하지 않으므로,[(다)단락에도 제시된 내용]** 선후 순서가 바뀌어야 되는 연산자들만 다른 위치로 이동하면 된다. 스택 알고리즘에서 피연산자는 순서가 되면 후위 표기로 바로 나오게 된다. 그러나 연산자의 경우 피연산자와 달리 뒤에 스택으로 들어갈 연산자의 존재 여부나 우선순위에 따라 스택에 보관되거나 후위 표기로 나오는 것이 결정된다. 이때 연산자의 우선순위는 괄호 안의 연산을 우선하되, 덧셈과 뺄셈보다 곱셈과 나눗셈을 우선하는 것이 원칙이다.**[후위 표기 식에서의 연산자들의 우선순위 결정 방법]** 예를 들어 컴퓨터에서 스택 알고리즘을 활용해 중위 표기 식 'A+B\*C'를 후위 표기 식으로 바꾸는 경우, 들어가는 데이터는 'A', '+', 'B', '\*', 'C'로 모두 5개이다. 들어가는 5개의 데이터 중 피연산자들은**[피연산자들 : A, B, C]** 스택에 보관되지 않고 후위 표기에 순차적으로 나오고, 연산자들은**[연산자들 : +, \*]** 스택에 순차적으로 보관된다. 다만 연산자들이 스택에 보관되거나 후위 표기로 나올 때에는 현재의 스택 톱에 있는 연산자와 우선순위가 비교되는 과정을 먼저 거친다. 만일 스택 톱에 있는 연산자의 우선순위가 보관되려는 연산자의 우선순위보다 높거나 같으면 스택 톱에 있는 연산자가 먼저 스택에서 나오게 된다. 그런 뒤에는 다시 스택 톱에 오게 되는 다른 연산자와 우선순위가 비교된다. 중위 표기식 다 읽고 나면 연산자가 존재하지 않아 스택에 남아 있는 모든 연산자가 톱에서부터 하나씩 나오게 된다.

스택 알고리즘을 활용해 중위 표기 식을 후위 표기 식으로 변환하는 방법

- ➡ 스택 알고리즘의 후위 표기 식을 활용할 경우 피연산자들의 순서는 변하지 않으나, 연산자들은 위치의 이동이 생긴다. -> 연산자들의 우선순위는 ①괄호 안의 연산이 먼저 ②덧셈과 뺄셈보다 곱셈과 나눗셈이 먼저.
- ➡ 연산자들이 스택에 보관되거나 후위 표기로 나올 경우에도 스택 톱에 있는 연산자와 우선순위를 비교하는 과정을 거쳐야 톱에서부터 하나씩 나올 수 있음.

(라) 단락의 독해 포인트 : 중위 표기 식을 후위 표기 식으로 변환할 때 연산자들의 우선순위 결정되는 방법과 과정을 예를 들어 설명함

**\*지문확인**

## 1단락

## (1) 스택의 특성과 스택 알고리즘의 개념

- 스택의 특성 : 한쪽 끝에서만 데이터가 들어가거나 나올 수 있음
- 스택 알고리즘의 개념 : 데이터가 들어가 보관되고 나오는 규칙

## 2단락

## (1) 스택 알고리즘의 쓰임

- 컴퓨터 내의 데이터 처리에서 수식 변환 과정에 활용됨

## (2) 수식의 세 가지 표기 식

- ① 중위 표기 식 : 연산자가 피연산자 가운데 있음 -> 일상생활에서 사용하는 표기 식
- ② 후위 표기 식 : 연산자가 피연산자 뒤에 있음 -> 컴퓨터 내부에서 활용
- ③ 전위 표기 식 : 연산자가 피연산자 앞에 있음

## 3단락

## (1) 중위 표기 식을 후위 표기 식으로 변환하는 방법

- ① 표기 식을 연산자 기준으로 괄호로 묶는다.
- ② 각 연산자를 묶고 있는 괄호의 오른쪽 괄호 밖으로 연산자를 이동한다.
- ③ 괄호를 모두 제거한다.

## 4단락

## (1) 스택 알고리즘을 활용해 중위 표기 식을 후위 표기 식으로 변환 할 때 피연산자와 연산자의 순서

- ① 피연산자들의 순서는 변하지 않는다.
- ② 연산자들은 뒤에 스택으로 들어갈 연산자의 존재 여부나 다른 연산자와의 우선순위 비교 과정을 통해 순서가 결정된다.

**\*지문 필수 어휘**

- 구조 : 부분이나 요소가 어떤 전체를 짜 이룸. 또는 그렇게 이루어진 일개.
- 활용하다 : 충분히 잘 이용하다.
- 변환되다 : 수식이나 함수, 관계식 등에서 제각기 바뀌어지다.
- 언급하다 : 어떤 문제에 대하여 말하다.
- 순차적 : 순서를 따라 차례대로 하는. 또는 그런 것.

**\*지문확인 문제**

1. 컴퓨터에서 데이터를 처리하기 위해 여러 가지 데이터 구조를 활용하는데 그 중 한쪽 끝에서만 데이터나 들어가거나 나올 수 있는 특성을 가진 것은?

---

2. 스택에서 데이터가 가장 먼저 나올 수 있는 위치는?

---

3. 스택의 구조는 ( )에, 튜브는 ( )에 비유될 수 있다.

---

4. 데이터가 들어가 보관되고 나오는 규칙을 무엇이라고 하는가?

---

5. 스택 알고리즘은 컴퓨터 내의 데이터 처리에서 ( )의 과정에서 활용된다.

---

6. 수식은 '+, \*, -, /' 등의 ( )와 'x, y, z, A, B, C' 등의 변수나 '1, 2, 3' 등의 상수를 포함하는 ( )로 구성된다.

---

7. 수식의 표기 식 중 일상생활에서 사용하는 것은 ( )이고, 이것을 컴퓨터 내부에서는 ( )으로 변환하여 계산한다.

---

8. 중위 표기 식을 후위 표기 식으로 변환할 때 ( 연산자 / 피연산자 )의 순서는 변하지 않는다.

---

9. 후위 표기 식으로 변환할 때 연산자를 묶고 있는 괄호의 ( 오른쪽 / 왼쪽 ) 밖으로 연산자를 이동한다.

---

10. 중위 표기 식을 전위 표기 식으로 변환할 때 연산자를 묶고 있는 괄호의 ( 오른쪽 / 왼쪽 ) 밖으로 연산자를 이동한다.

---

11. 스택 알고리즘에서 ( 연산자 / 피연산자 )는 순서가 되면 후위 표기로 바로 나오게 된다.

---

12. 스택 알고리즘을 활용하여 중위 표기 식을 후위 표기 식으로 변환할 때 연산자의 우선순위를 결정하는 원칙은?(2가지)

---

13. 스택 알고리즘에서 보관되려는 연산자의 우선순위보다 스택 톱에 있는 연산자의 우선순위가 ( ) 때, 스택 톱에 있는 연산자가 먼저 스택에서 나오게 된다.

---

**\*지문확인 문제 [서술형 대비]**

14. 후위 표기 식과 전위 표기 식의 차이점은?

---

15. 주어진 중위 표기 식 'A-B/C'를 후위 표기 식으로 변환하는 과정을 순서대로 서술하십시오.

---

16. 연산자들이 스택에 보관되거나 후위 표기로 나올 때 우선적으로 거쳐야 하는 과정은?

---

**\*연계 변형 문제**

※ 다음 글을 읽고, 물음에 답하시오.

(가) 컴퓨터에서는 데이터를 처리하기 위해 여러 가지 데이터 구조를 활용한다. 그 대표적인 것으로 ‘스택(stack)’을 들 수 있다. 스택은 한쪽 끝에서만 데이터가 들어가거나 나올 수 있는 특성을 가지는데, 스택에서 데이터가 가장 먼저 나올 수 있는 위치를 톱(top)이라 한다. 스택의 구조는 위쪽만 뚫려 있는 통 안에 책을 차곡차곡 쌓는다고 생각하면 쉽게 이해할 수 있다. 만약 n개의 책을 통에 쌓았다면 가장 먼저 꺼낼 수 있는 책은 아래에서부터 n번째에 있는 것이고, 그 위에 책을 또 올리게 되면 그 책은 아래에서부터 n+1번째에 있게 된다. 스택 구조는 이와 같은 통에, 톱은 가장 위에 있는 책의 위치에 비유될 수 있다. 그리고 스택 알고리즘은 데이터가 들어가 보관되고 나오는 규칙으로, 통에 책을 넣어 쌓고 책을 꺼내는 행위에 비유될 수 있다.

(나) 컴퓨터 내의 데이터 처리에서 스택 알고리즘은 수식 변환의 과정에 활용된다. 수식은 ‘+’(덧셈), ‘\*’(곱셈), ‘-’(뺄셈), ‘/’(나눗셈) 등과 같은 **①연산자**와, ‘x, y, z, A, B, C’ 등과 같은 변수나 ‘1, 2, 3’ 등과 같은 상수가 포함되는 **②피연산자**로 구성된다. 그리고 수식은 크게 세 가지 표기 식으로 표현할 수 있는데, 연산자가 피연산자 가운데 있는 **③중위 표기 식**, 연산자가 피연산자 앞에 있는 **④전위 표기 식**, 연산자가 피연산자 뒤에 있는 **⑤후위 표기 식**이 그것이다. 예를 들어 ‘x 더하기 y 나누기 z’는 중위 표기 식으로는 ‘x +y/z’로, 전위 표기 식으로는 ‘+x/yz’로, 후위 표기 식으로는 ‘xyz/ +’로 표현된다. 우리가 일상생활에서 사용하는 것은 중위 표기 식이지만, 컴퓨터 내부에서는 괄호를 사용하지 않고도 계산해야 할 순서를 알 수 있는 후위 표기 식을 활용하기 때문에 중위 표기 식으로 입력한 수식은 스택 알고리즘을 통해 후위 표기 식으로 변환되어 계산된다.

(다) 중위 표기 식을 후위 표기 식으로 변환하는 방법은 간단하다. 기본적인 전제는 피연산자들의 순서는 변하지 않는다는 것이다. 즉 중위 표기 식을 연산자 기준으로 괄호로 묶은 다음, 각 연산자를 묶고 있는 괄호의 오른쪽 괄호 밖으로 연산자를 이동한 후 괄호를 모두

제거하면 된다. 예를 들어 **③중위 표기 식 ‘A+B\*C’**를 후위 표기 식으로 바꾸기 위해서는 ‘(A +(B\*C))’로 괄호를 묶은 뒤, ‘(A(BC)\*)+’와 같이 괄호의 오른쪽에 연산자를 옮긴 후 ‘ABC\* +’와 같이 괄호를 제거하면 된다. 중위 표기 식을 전위 표기 식으로 변환하는 과정 역시 동일하다. 다만 연산자를 오른쪽에 옮기는 것이 아니라 왼쪽으로 옮기는 **①것**만 차이가 있다.

(라) 컴퓨터에서는 중위 표기 식을 후위 표기 식으로 변환하기 위해 앞에서 언급한 **②스택 알고리즘**을 활용한

다. 후위 표기 식에서 피연산자들의 순서는 변하지 않으므로, 선후 순서가 바뀌어야 되는 연산자들만 다른 위치로 이동하면 된다. 스택 알고리즘에서 피연산자는 순서가 되면 후위 표기로 바로 나오게 된다. 그러나 연산자의 경우 피연산자와 달리 뒤에 스택으로 들어갈 연산자의 존재 여부나 우선순위에 따라 스택에 보관되거나 후위 표기로 나오는 것이 결정된다. 이때 연산자의 우선순위는 괄호 안의 연산을 우선하되, 덧셈과 뺄셈보다 곱셈과 나눗셈을 우선하는 것이 원칙이다. 예를 들어 **③컴퓨터에서 스택 알고리즘을 활용해 중위 표기 식 ‘A+B\*C’를 후위 표기 식으로 바꾸는 경우**, 들어가는 데이터는 ‘A’, ‘+’, ‘B’, ‘\*’, ‘C’로 모두 5개이다. 들어가는 5개의 데이터 중 피연산자들은 스택에 보관되지 않고 후위 표기에 순차적으로 나오고, 연산자들은 스택에 순차적으로 보관된다. 다만 **④연산자들이 스택에 보관되거나 후위 표기로 나올 때에는 현재의 스택 톱에 있는 연산자와 우선순위가 비교되는 과정**을 먼저 거친다. 만일 스택 톱에 있는 연산자의 우선순위가 보관되려는 연산자의 우선순위보다 높거나 같으면 스택 톱에 있는 연산자가 먼저 스택에서 나오게 된다. 그런 뒤에는 다시 스택 톱에 오게 되는 다른 연산자와 우선순위가 비교된다. 중위 표기를 다 읽고 나면 연산자가 존재하지 않아 스택에 남아 있는 모든 연산자가 톱에서부터 하나씩 나오게 된다.

1. (가)~(라)의 각 문단에 제시된 내용을 정리한 것으로 옳지 않은 것은?

- ① (가) : 스택의 개념과 스택 알고리즘의 의미
- ② (나) : 수식 변환 과정에 활용되는 스택 알고리즘
- ③ (다) : 중위 표기 식을 후위 표기 식으로 변환하는 방법
- ④ (라) : 컴퓨터에서 중위 표기 식이 후위 표기 식으로 변환되는 과정
- ⑤ (라) : 스택 알고리즘에서 연산자들의 우선순위를 결정하는 기준

**[교재 변형]**

2. 밑줄에 대한 설명으로 가장 적절하지 않은 것은?

- ① 수식의 세 가지 표기 식으로 전위, 중위, 후위 표기 식을 제시하고 그 개념을 설명하고 있다.
- ② 스택 알고리즘이 활용되는 과정을 사례를 들어 소개하고 있다.
- ③ 중위 표기 식이 전위 표기 식으로 변환되는 과정을 보여주고 있다.
- ④ 스택과 스택 알고리즘의 특성에 대해 설명하고 있다.

⑤ 주요 소재를 적절한 상황에 빗대어 설명하고 있다.

3. 윗글을 이해한 내용으로 적절한 것은?

- ① '스택'은 한 쪽 끝에서 데이터가 들어가 반대편 끝 쪽으로 데이터가 나오는 특성을 가진다.
- ② n개의 책을 통해 쌓았을 때 가장 먼저 꺼낼 수 있는 책은 위에서부터 n번째에 있는 책으로, 이것은 스택에서 '톱'에 해당한다.
- ③ 연산자들이 스택에 들어갔다 나올 때에는 들어간 순서대로 나오게 된다.
- ④ 스택 알고리즘에서 변수나 상수의 순서는 변하지 않는다.
- ⑤ 중위 표기 식을 전위 표기 식으로 변환할 때에는 연산자를 괄호의 오른쪽으로 이동 후 괄호를 제거하면 된다.

4. ㉠에 대한 설명으로 적절하지 않은 것은?

- ① ㉠은 3개의 피연산자와 2개의 연산자로 구성되어 있다.
- ② ㉠을 연산자를 기준으로 괄호로 묶을 때에는 '\*'가 '+'보다 우선하는 원칙을 지켜 '(A+(B\*C))'로 묶는다.
- ③ ㉠을 'ABC\*+'로 변환한 것은 컴퓨터 내부에서는 괄호를 사용하지 않고도 계산 순서를 알 수 있는 식을 활용하기 때문이다.
- ④ ㉠을 전위 표기 식에 따라 변환할 때에는 후위 표기 식으로 변환할 때와 똑같이 '(A+(B\*C))'로 묶어 준다.
- ⑤ ㉠을 전위 표기 식으로 변환하면 '+\*ABC'이다.

※ [5~6][A]를 바탕으로 <보기>의 수식에 대한 물음에 답하십시오.

<보기>

$$A*(B+C+D)-E*F$$

5. <보기>의 수식을 후위 표기 식으로 변환한 것으로 옳은 것은?

- ① AB\*CD++EF-\*
- ② A\*BCD++EF-\*
- ③ ABCD++EF-\*\*
- ④ ABC+D+\*EF\*-
- ⑤ ABC+D+EF\*\*-

6. <보기>의 수식을 후위 표기 식으로의 변환하는 과정을 통해 알 수 있는 사실로 옳은 것은?

- ① 연산자들은 중위 표기 식에 표기된 순서대로 후위 표기 식으로 변환된다.
- ② 후위 표기 식으로 변환된 결과 연산자들은 먼저 연산이 진행되어야 하는 순서로 정렬되어 있다.
- ③ 후위 표기 식으로 변환되는 과정에서 피연산자들은 우선순위에 따라 순서가 변하게 된다.
- ④ 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선한다는 원칙에 의해 스택의 톱에는 항상 곱셈과 나눗셈이 존재한다.
- ⑤ 후위 표기 식으로 변환된 후 연산자를 기준으로 괄호로 묶어야 컴퓨터 내부에서 계산해야 할 순서를 알 수 있다.

**[교재 변형]**

7. 윗글을 근거로 할 때, <보기>의 빈 칸에 들어갈 표기 식으로 적절한 것은?

<보기>

전위 표기 식	⇐	중위 표기 식	⇒	후위 표기 식
$+*XY/ZW$		(㉠)		(㉡)

- |   | (㉠)       | (㉡)       |
|---|-----------|-----------|
| ① | $X*Y+Z/W$ | $XY*ZW/+$ |
| ② | $X+Y*Z/W$ | $XYZ*W/+$ |
| ③ | $X*Y/Z+W$ | $XY*/ZW+$ |
| ④ | $X*Y+Z/W$ | $XYZ**W/$ |
| ⑤ | $Y/Z*X+W$ | $YZ/XW+*$ |

8. ㉢에 대한 이해로 적절하지 않은 것은?

- ① 컴퓨터에서 데이터를 처리할 때 데이터가 들어가 보관되고 나오는 규칙을 말한다.
- ② 중위 표기를 후위 표기나 전위 표기로 변환하는 과정에 활용된다.
- ③ 우리가 일상에서 사용하는 표기 식은 ㉢을 통해 변환의 과정을 거친 것이다.
- ④ 컴퓨터 내부에서는 입력된 데이터를 ㉢을 통해 괄호를 사용하지 않고도 계산해야 할 순서를 알 수 있다.
- ⑤ 'A\*B-C+D'를 전위 표기 식으로 바꾸는 경우 들어가는 데이터는 모두 7개이다.

9. ‘스택 알고리즘’을 통한 변환 과정에 대한 이해로 적절하지 않은 것은?

- ① 중위 표기 식을 괄호로 묶을 때에는 연산자를 기준으로 하되, 우선순위를 정하는 원칙에 따른다.
- ② 피연산자들은 스택에 보관되지 않고 순서에 변화 없이 후위 표기로 바로 나온다.
- ③ 연산자들이 스택에 보관되거나 후위 표기로 나올 때에는 우선순위가 비교되는 과정을 먼저 거친다.
- ④ 중위 표기를 다 읽은 후에는 스택에 남아 있는 모든 연산자가 톱에서부터 하나씩 나오게 된다.
- ⑤ 스택 톱에 있는 연산자가 보관되려는 연산자보다 우선순위가 낮으면 보관되려는 연산자가 먼저 스택에서 나온 뒤 바로 뒤이어 나오게 된다.

10. ㉠과 ㉡에 대한 설명으로 옳은 것은?

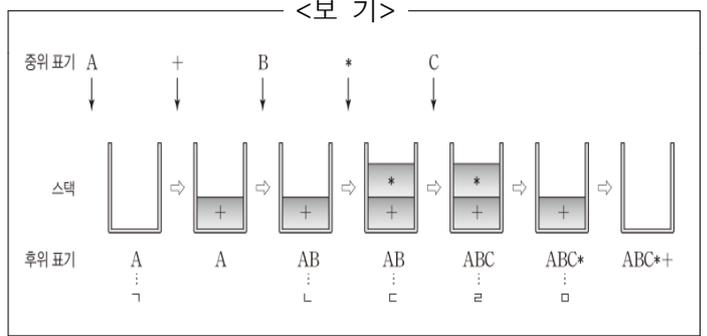
- ① 변수나 상수는 ㉠에, 사칙연산 부호는 ㉡에 해당한다.
- ② ㉠이 ㉡의 가운데에 있는 수식은 스택 알고리즘을 통해 ㉡의 앞이나 뒤로 변환된다.
- ③ ㉠나 ㉡만으로 수식은 구성되고 스택 알고리즘을 통해 변환될 수 있다.
- ④ ㉠과 ㉡는 모두 스택 알고리즘을 통한 변환 과정을 통해 선후 순서가 바뀌게 된다.
- ⑤ 스택 알고리즘을 통해 수식이 변환될 때에는 ㉠은 ㉠끼리, ㉡는 ㉡끼리 묶여서 스택에서 나오게 된다.

11. ㉢, ㉣, ㉤에 대한 설명으로 적절하지 않은 것은?

- ① ㉢는 우리가 일상생활에서 사용하는 것이지만, 컴퓨터 내부에서는 다른 표기로 변환되어 계산된다.
- ② ㉢를 ㉣로 변환할 때에는 연산자가 괄호의 왼쪽 밖으로 이동하게 된다.
- ③ ㉢를 ㉣나 ㉤로 변환할 때, <보기>는 우선 연산자를 계산해야 할 순서대로 괄호로 묶어줘야 한다.
- ④ ‘a/b+c’가 ㉢로 주어졌을 때, ㉣로 변환하면 ‘/+abc’가 된다.
- ⑤ ‘a/b+c’가 ㉢로 주어졌을 때, ㉤로 변환하면 ‘ab/c+’가 된다.

[교재 변형]

※[12-13] 다음 <보기>를 보고 물음에 답하시오.



12. <보기>는 ㉢의 과정을 도식화한 것이다. 윗글을 근거로 하여, ㉣~㉤에 대해 보인 반응으로 적절하지 않은 것은?

- ① ㉣에서 ‘A’는 피연산자이기 때문에 순서가 되어 후위 표기로 바로 나온 것이다.
- ② ㉣에서 ‘+’가 스택에 남아있는 것은 뒤에 피연산자가 더 존재하기 때문이라고 볼 수 있다.
- ③ ㉣에서 ‘\*’가 ‘+’보다 더 우선순위가 높기 때문에 스택의 톱에 위치하게 된 것이다.
- ④ ㉣에서 피연산자 ‘A, B, C’가 차례로 정렬되어 있는 것으로 피연산자들은 변환을 해도 순서에 변하지 않는다는 것을 알 수 있다.
- ⑤ ㉤에서 ‘\*’가 ‘+’보다 먼저 나온 것으로 보아 컴퓨터 내부에서는 ‘\*’를 우선 계산을 할 것이다.

13. <보기>의 중위 표기 식과 후위 표기 식에 대한 설명으로 적절한 것은?

- ① <보기>에 주어진 표기 식은 5개의 데이터로 구성되어 있으며, 그 중 연산자는 3개이다.
- ② <보기>의 중위 표기 식에서 ‘A+B’가 괄호로 묶여 있다 하더라도 후위 표기 식으로 변환된 값이 같을 것이다.
- ③ <보기>의 중위 표기 식에서 ‘\*’가 ‘/’으로 바뀐다면 ‘+’보다 연산 우선순위가 밀리게 될 것이다.
- ④ <보기>의 변환된 결과를 통해 중위 표기 식을 후위 표기 식으로 변환할 때에는 피연산자가 모두 나와야 연산자가 나올 수 있다는 것을 알 수 있다.
- ⑤ <보기>의 변환된 결과를 통해 컴퓨터 내부에서는 ‘B’와 ‘C’를 먼저 곱한 후 그 값에 ‘A’를 더하는 계산을 수행할 것이다.

14. (라)문단을 바탕으로 <보기>의 중위 표기 식을 후위 표기 식으로 변환하는 과정에 대한 이해로 적절하지 않은 것은?

<보 기>  
A+(B-C)\*D

- ① <보기>를 후위 표기 식으로 변환할 때 우선순위가 제일 높은 연산자는 '\*'이다.
- ② <보기>의 피연산자 'A, B, C, D'는 후위 표기 식으로 변환된 후에도 순서에 변함이 없다.
- ③ <보기>에서 '-'는 괄호 안에 있으므로 스택에 먼저 보관된 '+'보다 우선순위가 높아 스택 톱에 위치하게 된다.
- ④ <보기>를 후위 표기 식으로 변환하면 'ABC-D\*+'이다.
- ⑤ <보기>를 후위 표기 식으로 변환한 결과 나온 연산자들의 순서는 계산 순서와 일치한다.

[교재 변형]

※[15-16] 다음 <보기1>을 읽고 물음에 답하십시오.

<보 기 1>  
'큐(queue)'는 스택과 마찬가지로 컴퓨터에서 사용되는 데이터 구조 중 하나이다. 큐는 양쪽 끝만 터져 있는 선형 구조를 하고 있다고 볼 수 있는데, 한쪽 끝에서는 데이터가 들어가기만 하고, 다른 한쪽 끝에서는 들어간 순서대로 보관된 데이터가 나오기만 하는 특성을 나타낸다. 데이터가 들어가는 쪽은 '리어(rear)', 보관된 데이터가 나오는 쪽은 '프런트(front)'라고 한다.

15. 윗글의 '스택'과 <보기1>의 '큐'에 대한 설명으로 옳은 것을 <보기2>에서 모두 고른 것은?

<보 기 2>

ㄱ. '스택'과 '큐' 모두 컴퓨터에서 데이터가 들어가 보관되고 나오는 데이터 구조이다.

ㄴ. '스택'은 '큐'와 달리 한쪽 끝에서만 데이터가 들어가고 나온다.

ㄷ. '큐'는 스택과 달리 피연산자들과 연산자들의 순서에 모두 변화가 있다.

ㄹ. 같은 표기 식도 '스택'이나 '큐'냐에 따라 데이터가 나오는 순서가 다르다.

- ① ㄱ, ㄴ            ② ㄱ, ㄷ            ③ ㄱ, ㄴ, ㄹ
- ④ ㄱ, ㄷ, ㄹ      ⑤ ㄴ, ㄷ, ㄹ

16. '스택'과 '큐'를 비교한 내용으로 적절하지 않은 것은?

- ① '스택'은 '큐'와 달리 데이터가 들어가고 나오는 곳인 한 방향으로 제한되어 있다.
- ② '스택'은 '큐'와 달리 연산자가 들어간 순서와 나오는 순서가 다르다.
- ③ '스택'과 '큐' 모두 마지막에 저장된 데이터를 가장 먼저 꺼내는 구조이다.
- ④ '큐'는 놀이 공원에 입장할 때 앞에서부터 줄을 선 순서대로 입장하는 것과 같은 원리이다.
- ⑤ '큐'는 데이터가 들어가고 나오는 곳이 양쪽 끝에 위치한 선형 구조이다.

17. ㉠을 이해한 내용으로 옳은 것은?

- ① 현재의 스택 톱에 위치한 연산자가 보관되려는 연산자보다 먼저 들어왔으므로, 우선순위가 높다.
- ② 스택 톱에 위치한 연산자가 보관되려는 연산자보다 우선순위가 낮을 경우 스택을 나와서 보관되려는 연산자를 스택 톱에 위치하게 한다.
- ③ 스택 톱에 위치한 연산자가 '\*', 보관되려는 연산자가 '+'일 경우, 우선순위를 정하는 원칙에 따라 '+'가 스택 톱에 위치하게 된다.
- ④ 스택 톱에 위치한 연산자가 '+', 보관되려는 연산자가 '-'일 경우, '-'가 먼저 스택에서 나오게 된다.
- ⑤ 스택 톱에 위치한 연산자가 '/'일 경우, 보관되려는 연산자가 괄호 안의 연산이라면 우선순위가 밀리게 된다.

18. <보기>를 바탕으로 밑줄 친 부분이 ㉠의 문법적 성격과 다른 것은?

<보 기>

- 조사 : 체언이나 부사, 어미 따위에 붙어 그 말과 다른 말과의 문법적 관계를 표시하거나 그 말의 뜻을 도와주는 품사.
- 의존 명사 : 의미가 형식적이어서 다른 말 아래에 기대어 쓰이는 명사.

- ① 나에게는 휴식이 필요할 뿐이다.
- ② 해야 할 일이 많아 늦은 밤에도 잠들 수 없었다.
- ③ 나는 해야 할 말을 뒤로 한 채 자리에서 일어났다.
- ④ 나는 너만 좋아한다.
- ⑤ 그가 떠난 지 1년 만에 돌아왔다.

## 정답 및 해설

### \*지문확인 문제

1. <답> 스택
2. <답> 톱
3. <답> 톱, 가장 위에 있는 책의 위치
4. <답> 스택 알고리즘
5. <답> 수식의 변환
6. <답> 연산자, 피연산자
7. <답> 중위 표기 식, 후위 표기 식
8. <답> 피연산자
9. <답> 오른쪽
10. <답> 왼쪽
11. <답> 피연산자
12. <답> 괄호 안의 연산을 우선한다.  
덧셈과 뺄셈보다 곱셈과 나눗셈을 우선한다.
13. <답> 높거나 같을
14. <답> 후위 표기 식은 연산자를 괄호의 오른쪽으로 옮기고 전위 표기 식은 연산자를 괄호의 왼쪽으로 옮긴다.
15. <답>  $(A-(B/C)) \Rightarrow (A(BC)/)- \Rightarrow ABC/-$
16. <답> 현재의 스택 톱에 있는 연산자와 우선순위가 비교되는 과정을 먼저 거친다.

### \*연계 변형 문제

1	①	2	③	3	④	4	⑤	5	④
6	②	7	①	8	③	9	⑤	10	②
11	④	12	②	13	⑤	14	①	15	③
16	③	17	⑤	18	④	수고하셨습니다.			

#### 1. <답>

(가)단락에는 스택과 스택 알고리즘의 특성을 설명하고 있다. 스택의 개념이나 스택 알고리즘을 활용했을 때의 의미는 글 전체에 제시되어 있지 않다.

#### 오답풀이

4번 : 괄호 안의 연산을 우선으로 하고, 덧셈과 뺄셈보다는 곱셈과 나눗셈을 우선으로 한다고 제시되어 있다.

#### 2. <답> ③

(라)단락에 중위 표기 식이 후위 표기 식으로 변환되는 과정을 보여주고 있다. 전위 표기 식이 변환되는 과정은 드러나지 않는다.

#### 오답풀이

1번 : (나)단락

2번 : (라)단락

4, 5번 : (가)단락

#### 3. <답> ④

(나)단락에 의하면 변수나 상수는 피연산자에 해당하며 (라)단락에 후위 표기 식에서 피연산자들의 순서는 변하지 않는다고 제시되어 있다.

#### 오답풀이

1번 : (가)단락에 의하면 스택은 한쪽 끝에서만 데이터가 들어가거나 나올 수 있는 특성을 가진다.

2번 : (가)단락에 의하면 n개의 책을 통에 쌓았을 때 가장 먼저 꺼낼 수 있는 책은 아래에서부터 n번째에 있는 책이라고 했으므로 틀린 설명.

3번 : (라)단락에 의하면 연산자들이 스택에 보관되거나 후위 표기로 나올 때에는 현재의 스택 톱에 있는 연산자와 우선순위가 비교되는 과정을 거쳐 더 우위에 있는 것이 먼저 나오게 된다.

5번 : (나)단락에 의하면 연산자를 괄호의 오른쪽으로 이동하는 것은 후위 표기 식이다.

4. <답> ⑤

(다)단락에 의하면 전위 표기 식으로 변환할 때에는 괄호의 왼쪽으로 연산자를 옮겨야 한다.

따라서 순서대로 변환해보면 '(A+(B\*C)) ⇨ +(A\*(BC)) ⇨ +A\*BC'가 된다.

오답풀이

1번 : (나)단락 설명에 따라 연산자는 '+, \*'로 2개 피연산자는 'A, B, C'로 3개이다.

2번 : (라)단락에 의하면 덧셈과 뺄셈보다 곱셈과 나눗셈이 우선한다.

3번 : (나)단락에 의하면 중위 표기 식으로 입력된 수식이 컴퓨터 내부에서는 괄호를 사용하지 않고도 계산 순서를 알 수 있는 후위 표기 식으로 변환된다.

④ : (다)단락에 의하면 후위 표기 식과 전위 표기 식의 변환하는 과정은 동일하고 다만 연산자를 괄호의 어느 쪽으로 옮기냐의 차이가 있다.

5. <답> ④

주어진 수식을 괄호로 먼저 묶은 후 괄호의 오른쪽 밖으로 연산자들을 옮기면 된다. 이 때 주의할 점은 원칙대로면 '\*'가 '+'보다 우선하긴 하지만 주어진 수식에서 이미 (B+C+D)가 괄호로 묶여있으니 (라)단락에 따라 괄호 안의 연산이 '\*'보다 우선한다.

$$(A*(B+C+D))-(E*F) \Rightarrow (A*(BC+D+))-(EF*) \Rightarrow (A(BC+D+))*(EF*)- \Rightarrow ABC+D+*EF*-$$

6. <답> ②

(나)단락에 후위 표기 식은 괄호를 사용하지 않고도 계산해야 할 순서를 알 수 있다고 했으므로 연산자들이 먼저 연산이 진행되어야 하는 순서로 정렬되어 있음을 알 수 있다.

오답풀이

1번 : (라)단락의 내용에 따르면 중위 표기 식의 연산자들이 변환을 통해 연산 순서대로 배열된다.

3번 : (라)단락을 보면 피연산자들의 순서는 변하지 않는다고 했다.

4번 : (라)단락을 보면 연산자의 우선순위는 괄호 안의 연산을 우선한다고 했으므로 스택의 톱에 항상 곱셈과 나눗셈이 우선하는 것은 아니다.

5번 : (나)단락을 보면 컴퓨터 내부에서는 괄호를 사용하지 않고도 계산해야 할 순서를 알 수 있는 후위 표기 식을 활용한다고 했으므로 괄호를 하지 않아도 계산 순서를 알 수 있다.

7. <답> ①

<보기>에 주어진 전위 표기를 우선 중위 표기 식으로 변환 후 다시 후위 표기 식으로 변환해야 한다.

$$\begin{aligned} <\text{전위 표기 식}> \text{+*XY/ZW} &\Rightarrow <\text{중위 표기 식}> X*Y+Z/W \\ &\Rightarrow ((X*Y)+(Z/W)) \Rightarrow ((XY*)+(ZW/)) \Rightarrow ((XY*)(ZW/))+ \\ &\Rightarrow <\text{후위 표기 식}> XY*ZW/+ \end{aligned}$$

8. <답> ③

(나)단락 설명에 의하면 우리가 일상생활에서 사용하는 것은 중위 표기 식이므로 스택 알고리즘이 활용되지 않은 상태이다.

오답풀이

1번 : (가)단락

2번 : (나)단락. 스택 알고리즘은 수식 변환의 과정에 활용된다.

4번 : (나)단락. 스택 알고리즘을 통해 중위 표기 식이 후위 표기 식으로 변환되어 괄호가 없어도 계산해야 할 순서를 알 수 있다.

5번 : (라)단락에 주어진 예 'A+B\*C'의 데이터는 연산자와 피연산자를 모두 합하여 5개. 선지에 있는 데이터도 연산자 'A, B, C, D'와 피연산자 '\*', '+, -'를 합해서 7개이다.

9. <답> ⑤

(라)단락을 보면 스택 톱에 있는 연산자의 우선순위가 보관되려는 연산자의 우선순위보다 높거나 같으면 스택 톱에 있는 연산자가 먼저 스택에서 나오게 된다. 그런데 만약 스택 톱에 있는 연산자가 보관되려는 연산자의 우선순위보다 낮다면 보관되려는 연산자가 먼저 스택에서 나오게 될 것이다. 그럼 남아 있는 연산자는 다시 보관되려는 연산자와의 우선순위 비교를 통해 순위가 높거나 같으면 스택에서 나올 것이고, 순위가 낮으면 스택에서 나오는 순위가 또 밀릴 것이다.

오답풀이

1번 : (다)단락

2, 3, 4번 : (라)단락

10. <답> ②

(나)단락. ㉠연산자가 ㉢피연산자 가운데 있는 중위 표기 식은 스택 알고리즘을 통해 전위 표기 식이나 후위 표기 식으로 변환된다.

오답풀이

1번 : (나)단락. 변수나 상수는 피연산자에, 사칙연산 부호는 연산자에 해당한다.

3번 : (나)단락. 연산자와 피연산자가 함께 수식을 구성한다.

4번 : (라)단락. 수식 변환 과정에서 선후 순서가 바뀌는 것은 연산자만이다. 피연산자의 순서는 변하지 않는다.

5번 : 수식이 변환될 때에는 어떤 식이 주어지냐에 따라 연산자와 피연산자가 뒤섞여 나올 수도 있다.

EX)  $A+B+C \Rightarrow AB+C+$

11. <답> ④

중위 표기 식 'a/b+c'를 전위 표기 식으로 변환하면 '(a/b)+c'  $\Rightarrow$  '(/ab)+c'  $\Rightarrow$  '(+/ab)c'  $\Rightarrow$  '+/abc'가 된다.

오답풀이

1번 : (나)단락. ㉠중위 표기 식은 우리 일상에서 사용하지만 컴퓨터 내부에서는 스택 알고리즘을 통해 후위 표기 식으로 변환되어 계산된다.

2번 : (다)단락. ㉠중위 표기 식을 ㉡전위 표기 식으로 변환할 때에는 연산자를 괄호의 왼쪽 밖으로 옮긴다.

3번 : (다)단락. 중위 표기 식을 전위 표기 식이나 후위 표기 식으로 변환할 때에는 연산자를 기준으로 괄호로 묶는다. 또한 (라)단락을 보면 연산자의 우선순위는 괄호 안의 연산이 먼저이고, 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선이다. 즉 계산 순서대로 괄호를 묶어주는 것이다.

5번 : 'a/b+c'를 ㉢후위 표기 식으로 변환하면 '(a/b)+c'  $\Rightarrow$  '(ab/)+c'  $\Rightarrow$  '(+ab/)+c'  $\Rightarrow$  'ab/c+'가 된다.

12. <답> ②

ㄴ에서 '+'가 스택에 남아있는 것은 뒤에 피연산자가 아니라 스택으로 들어갈 연산자가 더 존재하기 때문이다.

오답풀이

1번 : (라)단락에 의하면 피연산자는 순서가 되면 바로 나오기 때문에 스택에 보관되지 않는다.

3번 : (라)단락을 보면 스택 톱에 있는 연산자가 먼저 스택에서 나오게 되는데 이 때 스택 톱에 있는 연산자와 보관되려는 연산자의 우선순위 비교 과정을 거쳐 우선순위가 더 높은 연산자가 스택 톱에 위치하게 된다.

4번 : (라)단락을 보면 후위 표기 식에서 피연산자들의 순서는 변하지 않는다.

5번 : (라)단락을 보면 스택 톱에 있는 연산자가 먼저 스택에서 나오게 되는 것이고 이것은 스택 톱에 있는 연산자가 계산에서의 우선순위가 높아 먼저 계산된다는

것을 알 수 있다.

13. <답> ⑤

연산자의 우선순위가 높은 연산자부터 스택에서 나오게 되므로 'B\*C'가 먼저 계산되고 그 후 그 값에 'A'가 더해질 것이다.

오답풀이

1번 : 연산자는 '+, \*' 2개이다.

2번 : (라)단락을 보면 괄호 안의 연산을 우선한다고 했기 때문에 'A+B'가 괄호로 묶여있다면 그 연산을 먼저 진행해야 하므로 후위 표기 식 값이 달라질 것이다.

3번 : (라)단락을 보면 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선한다고 했으므로 '/'로 바뀐다 하더라도 '+'보다 우선 계산될 것이다.

4번 : <보기>는 피연산자가 먼저 나온 후 연산자가 나오는 결과를 보여주는 하지만, 어떤 연산자가 오고 괄호가 어디에 묶여 있냐에 따라 피연산자와 연산자가 뒤섞여 정렬될 수도 있다.

14. <답> ①

(라)단락을 보면 괄호 안의 연산을 우선한다고 하였으므로 우선순위가 제일 높은 연산자는 '-'이다.

오답풀이

2번 : (라)단락을 보면 피연산자들의 순서는 변하지 않는다.

3번 : (라)단락을 보면 괄호 안의 연산을 우선한다고 하였으므로 '-'는 가장 먼저 나올 수 있는 스택 톱에 위치한다.

4번 : 후위 표기 식으로 변환하는 과정을 살펴보면  $(A+((B-C)*D)) \Rightarrow (A+((BC-)*D)) \Rightarrow (A+((BC-)*D*)) \Rightarrow (A+((BC-)*D*))+ \Rightarrow ABC-D*+$

5번 : 변환된 결과의 연산자들 순서는 '-', \*, '+'이고, 실제로 계산하는 순서와 일치함을 알 수 있다.

15. <답> ③

ㄱ. <보기1>과 (가)단락을 보면 '스택'과 '큐' 모두 데이터가 보관되고 나가는 데이터 구조라는 것을 알 수 있다.

ㄴ. (가)단락을 보면 '스택'은 데이터가 들어가고 나오는 것이 한쪽 끝에서만 이루어 진다.

ㄷ. 윗글을 통해 '스택'은 연산 순서대로 데이터가 나온다는 것을 알 수 있고, <보기1>을 통해 '큐'는 데이터가 들어간 순서대로 데이터가 나온다는 것을 알 수 있다.

오답풀이

ㄷ. '큐'는 데이터가 들어간 순서대로 보관되고 나오기 때문에 연산자나 피연산자의 순서에 변화가 없다.

16. <답> ③

(가)단락에 통해 책을 쌓는 비유를 통해 '스택'은 가장 나중에 들어간 데이터가 먼저 나온다는 것을 알 수 있다. 그러나 '큐'는 <보기1>을 보면 들어간 순서대로 나온다.

오답풀이

1번 : (가)단락. '스택'은 한쪽 끝에서만 데이터가 들어가서 나온다.

2번 : 윗글을 보면 '스택'은 수식 변환 과정을 통해 가장 먼저 연산이 이루어져야 하는 연산자부터 나오게 된다.

4번 : <보기1>을 보면 '큐'는 데이터가 들어간 순서대로 나온다고 했으므로, 놀이공원 앞에서 줄을 선 순서대로 입장하는 것과 같은 원리이다.

5번 : <보기1>을 보면 '큐'에서는 데이터가 양쪽 끝에서만 들어가고 나오는 선형 구조이다.

17. <답> ⑤

(라)단락을 보면 연산자의 우선순위는 괄호 안의 연산을 우선한다고 했으므로 보관되려는 연산자가 괄호 안의 연산이라면 우선순위가 제일 높아져 스택톱에 위치하게 된다.

오답풀이

1번 : (라)단락을 보면 연산자가 들어온 순서로 우선순위가 정해지는 것은 아니다.

2번 : (라)단락을 통해 스택 톱에 위치한 연산자가 보관되려는 연산자보다 우선순위가 낮을 경우 나오지 못하고 스택 톱의 아래에 위치하게 된다는 것을 알 수 있다.

3번 : (라)단락에 곱셈과 나눗셈이 덧셈과 뺄셈보다 우선순위가 더 높으므로, 스택 톱에 위치한 연산자가 '\*', 보관되려는 연산자가 '+'일 경우, 우선순위를 정하는 원칙에 따라 '\*'가 스택 톱에 위치하게 된다.

4번 : (라)단락을 보면 스택 톱에 위치한 연산자가 보관되려는 연산자보다 우선순위가 높거나 같으면 먼저 스택에서 나온다고 했다. '+'와 '-'는 우선순위가 같으므로 먼저 스택 톱에 위치해 있던 '+'가 먼저 스택에서 나오게 된다.

18. <답> ④

㉑의 '것'은 의존명사이다. ④의 '만'은 조사로, 조사는 앞말과 붙여 쓴다.

오답풀이

1, 2, 3, 5번 : 모두 의존명사. 의존 명사는 앞말과 띄어쓰며 반드시 앞에 관형어가 존재한다.