

# 1. 개요

Java나 C에서 제공하는 기본 자료형을 쓰면 제한된 크기의 수만을 다룰 수 있습니다. (예 : 32 bits를 사용하는 int형 변수는 -2147483648 ~ 2147483647 범위의 값을 가집니다.) 이번 과제에서는 매우 큰 정수에 대해서도 제한없이 연산을 할 수 있도록 기본 자료형을 쓰지 않고 임의의 크기를 갖는 수를 char 배열(또는 byte 배열)로 받아 +, -, \* 연산을 수행하는 프로그램을 구현합니다. 가장 기본이 되는 자료구조인 배열의 개념을 이해하고 활용하는 것이 과제의 목적입니다.

## 2. 뼈대 코드

이 코드를 기본으로 하여 내용을 추가하도록 합니다. 이 코드에는 자료구조 숙제를 하기 위한 여러가지 기초 테크닉들이 구현되어 있으므로 단순히 사용하지만 말고 확실히 분석해 자기 것으로 만들기 바랍니다. 필요한 경우에만 이 코드를 전부, 혹은 부분적으로 참조하면 되며 이 코드와 전혀 다른 방식으로 숙제를 해도 좋습니다.

## 3. 입출력 형식

주어진 소스 코드에 입력 방법이 구현되어 있으므로 분석해 보기 바랍니다.

0 프로그램을 실행하면 한줄로 된 식을 입력받습니다.

1 다음으로 "A+B" 형태의 문자열을 입력받아 A와 B 두 값을 각각 char 배열(또는 byte 배열)로 저장합니다.

2 연산을 수행한 뒤 그 결과를 출력합니다. (연산자는 +, -, \* 만 씁니다.)

3 한 식의 계산이 끝나면 다음 식을 입력받습니다. (quit를 입력받으면 프로그램을 종료합니다.)

```

$ java BigInteger      ← 프로그램 실행
1000000000000000000+2000000000000000000 ← 이렇게 입력하면
21000000000000000000      ←이렇게 출력됨

2000000000000000000-1000000000000000000 ← 이렇게 입력하면
-8000000000000000000      ← 이렇게 출력됨

3000000000000000000 - 2000000000000000000 ← 이렇게 입력하면
-1700000000000000000      ← 이렇게 출력됨

500000000 *+1000 ← 이렇게 입력하면
500000000000      ← 이렇게 출력됨

-10000000 + 0      ←이렇게 입력하면
-10000000          ← 이렇게 출력됨

quit      ← 이렇게 입력하면
$          ← 종료한다.

```

## 4. 식의 표현

- ⑩ 입력받는 각 수의 크기는 최대 100자리입니다. 따라서 출력은 +, -인 경우에는 101자리, \*인 경우에는 200자리까지 될 수 있습니다.
- ⑩ 식에서 각 수의 부호는 +일 수도 있고 -일 수도 있고 아무것도 없을 수도 있습니다. 아무것도 없을 때는 +입니다.
- ⑩  $1+-2$  와 같은 경우는  $1+(-2)$ 로 계산합니다. 즉, 음수를 먼저 취한 다음 덧셈을 합니다.
- ⑩ 수식상에서 공백은 0개 이상 들어갈 수 있습니다. 숫자의 앞뒤와 연산자 사이에 공백이 들어갈 수 있다는 뜻입니다.(공백은 “ ”만 포함합니다.)

## 5. 유의사항

⑩ 프롬프트 (>)를 출력할 필요는 없습니다. 입력만 잘 받으면 됩니다. 프롬프트를 출력할 경우 출력결과가 이상해질 수도 있습니다.

⑩ 한 번에 연산자 하나만 처리하면 됩니다. 즉,  $A+B$ ,  $A*B$ ,  $A-B$  같은 것만 처리하면 되고,  $A+B+C$  같이 연산자가 두 개 이상 있는 경우는 처리하지 않아도 됩니다.

⑩ 올바른 입력만 들어온다고 가정하셔도 좋습니다.

⑩ 각 자리의 숫자들을 저장하기 위한 자료구조로는 primitive type(char, byte, int, ...)의 배열만 사용합니다. ArrayList, Vector, LinkedList 등 이미 만들어져 있는 것은 사용하면 안 됩니다.

⑩ 제출 전, 반드시 컴파일 가능 유무를 확인합니다. 컴파일시 옵션이 옵션은 쓰지 않습니다.

⑩ 아래와 같은 명령어를 입력하면 컴파일이 이루어져야 하며, BigInteger 라는 이름의 클래스가 생성되어야 채점이 이루어집니다.

```
$ javac BigInteger.java
```

⑩ 컴파일 후, 일정 시간 초과시 프로그램을 강제 종료시키기 위해 다음과 같이 timeout 명령어를 반드시 사용합니다.

```
$ timeout [수행시간(초)] java BigInteger  
    timeout 0.5 java BigInteger // 0.5초 수행  
    timeout 1 java BigInteger // 1초 수행
```

⑩ 실행시간이 testset 기준으로 가상서버에서 40초를 초과할 시 감점이 있습니다. testset은 채점 시 새로 생성하므로 주어진 testset 실행시간과 다를 수 있습니다. 따라서 여유있게 완료하는 것을 권합니다.

## 6. 자주 묻는 질문(필독!!!)

### 1) 내부 자료 저장 형식에 관하여

문제에 제시된 조건은 '각 자리의 숫자로 저장하기 위한 자료구조로는 primitive type의 배열만 사용합니다'입니다.

한 자리의 정수를 저장하는 방식으로는 byte 배열을 사용하는 것을 추천하나, int 배열을 사용해도 별다른 불이익은 없도록 하겠습니다.

### 2) 입력하는 수식의 형식

테스트 및 채점에 사용되는 수식은 아래의 형식을 따릅니다. 이외의 입력은 채점의 대상이 되지 않습니다.

(0개 이상의 공백) (부호 0~1개) (0개 이상의 공백) (숫자) (0개 이상의 공백) (+/- /\*) (0개 이상의 공백) (부호 0~1개) (0개 이상의 공백) (숫자) (0개 이상의 공백) 숫자의 시작은 1~9의 숫자입니다. 0001 같은 입력은 없다고 가정하시면 됩니다.

### 3) 채점 방식

입력 파일과 출력 파일이 정의되어 있으며, 입력 파일대로 프로그램에 입력했을 때 출력 파일과 프로그램의 출력 결과가 완전히 일치하는 경우에만 올바른 결과로 인정됩니다.

아래와 같은 경우, 올바른 결과가 아니니 주의하시기 바랍니다.

- 부가적인 출력을 하는 경우

(예: '수식을 입력하세요.'를 출력하는 경우)

- 디버깅용 출력을 지우지 않은 경우
- 눈에 보이지 않는 문자를 출력하는 경우 (char 배열의 데이터를 그대로 출력하는 경우 이런 문제가 발생할 수 있습니다.)
- 결과값이 양수일 때 '+'를 출력하는 경우