

과제3 (40점)

다음의 EBNF로 문법이 정의되는 언어를 위한 Parser를 구현하시오. [C, C++, Java, Python 중에서 하나 선택하여 구현]

[1] EBNF(1)

```
<program> → {<statement>}
<statement> → <var> = <aexpr> ; |
               print <aexpr> ;
<aexpr> → <term> {+ <term> | - <term>}
<term> → <number> | <var>
<number> → <dec> {<dec>}
<dec> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<var> → a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t
        | u | v | w | x | y | z
```

[2] EBNF(2)

```
<program> → {<declaration>} {<statement>}
<declaration> → <type> <var> ;
<statement> → <var> = <aexpr> ; |
               print <aexpr> ; |
               while (<bexpr>) ' { ' {<statement>} ' } '
<bexpr> → <aexpr> <relop> <aexpr>
<relop> → == | != | < | > | <= | >=
<aexpr> → <term> {+ <term> | - <term>}
<term> → <number> | <var>
<number> → <dec> {<dec>}
<type> → int
<dec> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<var> → a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t
        | u | v | w | x | y | z
```

▶ 입력

- 프로그램을 실행하면 사용자는 바로 코드를 입력할 수 있도록 구현함. (엔터키 입력까지를 하나의 프로그램으로 인식함)

▶ 출력

- 문법에 맞는 코드가 입력된 경우에는 다음 코드를 입력 받음
- 문법에 맞는 코드가 입력된 경우, 출력할 결과가 있다면 출력을 수행함 (선택 구현: 구현방식(2) 또는 구현방식(3)으로 제출 시)
- 문법에 맞지 않는 코드가 입력된 경우에는 "Syntax Error!"를 출력한 후, 다음 코드를 입력 받음
- 연속해서 2번 이상 아무것도 입력되지 않은 경우에는 프로그램 수행을 종료함

▶ 구현 방식은 3가지로 가능함

- 구현방식(1): EBNF(1) Parser 구현, 대입 및 출력 기능 미구현
- 구현방식(2): EBNF(1) Parser 구현, 대입 및 출력 기능 구현
- 구현방식(3): EBNF(2) Parser 구현, 대입, 출력 및 while문 기능 구현

▶ 실행예

```
>> a = 5; b = 3;
>> ab = 12;
>> Syntax Error!
>> int k = 2;
>> Syntax Error!
>> x = 10 + 5 * 2;
>> Syntax Error!
>> x = 10 + 5 - 2; print x;
>> 구현방식(1)은 아무것도 출력하지 않음. 구현방식(2) 또는 구현방식(3)은 13 출력
>> int k; int j; k = 3; j = 20; print k + j
>> Syntax Error!
>> int k; int j; k = 3; j = 20; print k + j;
>> 구현방식(1) 또는 구현방식(2)는 Syntax Error! 출력, 구현방식(3)은 23 출력
>> k = 10; print k + 15;
>> 구현방식(1)은 아무것도 출력하지 않음. 구현방식(2) 또는 구현방식(3)은 25 출력
>> e = 10; p = 9; print k + 15;
>> 구현방식(1)은 아무것도 출력하지 않음. 구현방식(2) 또는 구현방식(3)은 15 출력 (선언되지 않은 var의 값은 0으로 가정하고 수행)
>> int k; int j; k = 3; j = 20; int a; print k + j;
>> 구현방식(1), 구현방식(2), 구현방식(3) 모두 Syntax Error! 출력
```

```
>> int k; int j; k = 3; j = 20; print k + p;
>> 구현방식(1) 또는 구현방식(2)는 Syntax Error! 출력, 구현방식(3)은 3 출력 (선언
되지 않은 var의 값은 0으로 가정하고 수행)
>> int k; int j; k = 3; j = 20; print k + j - 1;
>> 구현방식(1) 또는 구현방식(2)는 Syntax Error! 출력, 구현방식(3)은 22 출력
>> int i; int j; i = 0; j = 0; while (i < 5) { j = j + i; i = i + 1;} print i; print j;
>> 구현방식(1) 또는 구현방식(2)는 Syntax Error! 출력, 구현방식(3)은 5 10 출력
```

▶ 제출 요구사항

- 구현 및 테스트를 완료한 소스코드는 (C, C++, Java, Python 중 1개 언어 선택) 하나의 zip 파일로 압축하여 제출해야 함
- 보고서는 별도의 PDF 파일로 제출해야 함
- 보고서에는 본인이 구현한 코드를 개략적으로 설명해야 하고, 실행 결과 스크샷은 반드시 포함하여야 함. 보고서에 소스 코드 전체를 포함할 필요는 없음
- 구현방식(1)은 17점 만점, 구현방식(2)는 25점 만점, 구현방식(3)은 35점 만점으로 평가함. 보고서 점수 5점. 20점 이상 과제 Pass임
- 보고서 첫째줄에 구현방식을 기재해야 함 (구현방식 (1), (2), (3) 중에서 하나 반드시 기재)
- 제출 요구사항 미준수 시에는 10% 감점 처리함