

과제2 (30점)

다음의 EBNF로 문법이 정의되는 수식을 위한 계산기를(Recursive-Descent Parser 구현 포함) C/C++, Java, Python으로 각각 구현하시오. (각 언어별로 소스코드 파일 1개씩 총 3개의 소스코드 파일 제출)

```
<expr> → <bexpr> | <aexpr>
<bexpr> → <aexpr> <relop> <aexpr>
<relop> → == | != | < | > | <= | >=
<aexpr> → <term> { * <term> | / <term> }
<term> → <factor> { + <factor> | - <factor> }
<factor> → <number> | (<aexpr>)
<number> → <dec> {<dec>}
<dec> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

▶ 입력

- 프로그램을 실행하면 초기화면에서 사용자는 수식을 입력함. 엔터키가 입력되면 하나의 수식이 입력된 것으로 처리함

▶ 출력

- 문법에 맞는 수식이 입력된 경우에는 수식 결과를 계산하여 출력하고, 다음 수식을 입력 받음. <bexpr>의 결과는 true 또는 false이고, <aexpr>의 결과는 숫자임
- 문법에 맞지 않는 수식이 입력된 경우에는 "syntax error!!"를 출력하고, 다음 수식을 입력 받음
- 아무 문자도 입력되지 않은 경우에는 프로그램 수행을 종료함

▶ 실행예

```
>> ((12 + 3)
>> syntax error!!
>> (35 / 5) - 2 != 20 - 10 / 2
>> false
>> (32+      12    *7)
>> 308
>> (12 * 2) + 30    * 5
>> 270
```

▶ **제출 요구사항**

- 구현 및 테스트를 완료한 소스코드 파일 3개는 (C/C++, Java, Python 각각 1개씩) 하나의 파일로 압축하여 제출해야 함
- 보고서는 별도의 PDF 파일로 제출해야 함
- 보고서에는 본인이 구현한 코드를 개략적으로 설명해야 하고, 실행 결과 스냅샷은 반드시 포함하여야 함. 보고서에 소스 코드 전체를 포함할 필요는 없음
- <aexpr> 구현만 완료한 경우에는 15점을 부여하고, <bexpr>까지 모두 구현을 완료한 경우에는 25점을 부여함. 보고서 점수 5점. 18점 이상 과제 Pass임
- 제출 요구사항 미준수 시에는 10% 감점 처리함