symantics는 컴파일러한테 맡긴다.
parsing만 하면 된다.

## 구현과제1 (36점)

다음의 EBNF로 문법이 정의되는 언어를 위한 Recursive-Descent Parser를 C/C++, Java, Python으로 각각 구현하시오. (각 언어별로 소스코드 파일 1개씩 총 3개의 소스코드 파일 제출, 구현 언어별 과제 점수 12점씩, 총 36점)

**<program> → {<statement>}**
**<statement> → <var> = <expr> ; | print <var> ;**
**<expr> → <bexpr> | <aexpr>**
**<bexpr> → <number> <relop> <number>**
**<relop> → == | != | < | > | <= | >=**
**<aexpr> → <term> {( + | - ) <term>}**
**<term> → <factor> {( * | / ) <factor>}**
**<factor> → <number>**
**<number> → <dec> {<dec>}**
**<dec> → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9**
**<var> → x | y | z**

토큰은 다붙어있고 토큰과 토큰 사이에는 공백이 한개 이상 들어갈 수 있다.

▸ **입력**
- 프로그램을 실행하면 사용자는 바로 코드를 입력할 수 있도록 구현함. (엔터키 입력까지를 하나의 프로그램으로 인식함)
- 입력 코드의 토큰과 토큰 사이에는 공백 문자가 들어감
- 입력 코드의 <aexpr>의 결과값은 <number> 범위 숫자로 한정함

▸ **출력**
- 문법에 맞는 코드가 입력된 경우에는 다음 코드를 입력 받음
- 문법에 맞는 코드가 입력된 경우, 출력할 결과가 있다면 출력을 수행함. <bexpr>의 결과는 TRUE 또는 FALSE이고, <aexpr>의 결과는 숫자임
- 문법에 맞지 않는 코드가 입력된 경우에는 "syntax error!!"를 출력한 후, 다음 코드를 입력 받음
- terminate 가 입력된 경우에는 프로그램 수행을 종료함

값이 주어지지 않은 xyz를 출력해야하면 디폴트로 0을 출력해라.

▸ **실행예**

**>> x = ( 12 + 3 ;**
**>> syntax error!!**
**>> x = 12 == 3  print x ;**
**>> syntax error!!**

**>> y = 12 == 3 ; print a ;**
**>> syntax error!!**
**>> x = 12 == 3 ; print x ;**
**>> FALSE**
**>> z = 100 * 3 ; y = 42 * 5 ; y = 12 != 3 ; print y ;**
**>> TRUE**
**>> x = 12 == 3 ; y = 10 + 5 * 3 ; print y ; print x ;**
**>> 25 FALSE**
**>> y = 10 * 5 - 3 ; z = 5 – 2 + 8 / 2 ; print y ; print z ;**
**>> 47 7**

### ▸ 제출 요구사항

- 구현 및 테스트를 완료한 소스 코드 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
- 제출 요구사항 미준수 시에는 10% 감점 처리함