5) Ohm Grammer

A) -22 evaluates to 4**

```
Exp       = Exp ("+" | "-") Term          --binary
          | Term
Term      = Term ("*" | "/" | "%") Factor --binary
          | Factor
Factor    = Primary "**" Factor           --binary
          | Primary
Primary   = id
          | numeral
          | "(" Exp ")"                   --parens
```

B) -22 evaluates to -4**

```
Exp       = Exp ("+" | "-") Term          --binary
          | Term
Term      = Term ("*" | "/" | "%") Factor --binary
          | Factor
Factor    = Primary "**" Factor           --binary
          | Primary
Primary   = id
          | numeral
          | "-" Exp                       --unary
          | "(" Exp ")"                   --parens
```

C) -22 is a syntax error, while allowing (-2)**2 and -(2**2) to be legal**

```
Exp       = Exp ("+" | "-") Term          --binary
          | Term
Term      = Term ("*" | "/" | "%") Factor --binary
          | Factor
Factor    = Primary "**" Factor           --binary
          | "-" Primary                   --negation
          | Primary
Primary   = id
          | numeral
          | "-" Exp                       --unary
          | "(" Exp ")"                   --parens
```

6) Astro++

```
AstroPlusPlus {
Program       = Statement+
Statement     = id "=" Exp ";"                          --assign
              | print Exp ";"                           --print
              | while Exp "{" BStatement* "}"           --while
              | if Exp "{" Statement* "}" (elseIf Exp "{" Statement* "}")*
                (else "{" Statement* "}")?              --if

BStatement  = id "=" Exp ";"                            --assign
              | print Exp ";"                           --print
              | break ";"                               --break
              | while Exp "{" BStatement* "}"           --while
              | if Exp "{" BStatement* "}" (elseIf Exp "{" BStatement* "}")*
                (else "{" BStatement* "}")?             --if

Exp           = Condition binaryOp Condition            --binary
              | Condition

Condition     = Condition addOp Term                    --binary
              | Term

Term          = Term mulOp Factor                       --binary
              | Factor

Factor        = Primary "**" Factor                     --binary
              | "-" Primary                             --negation
              | Primary

  Primary     = id
              | number
              | "(" Exp ")"                             --parens

  keyword     = print | if | elseIf | else | while | break
  print       = "print" ~idchar
  if          = "if" ~idchar
  elseIf      = "elseIf" ~idchar
  else        = "else" ~idchar
  while       = "while" ~idchar
  break       = "break" ~idchar

  idchar      = letter | digit | "_"

  id          = ~keyword letter idchar*
```

```
  number        = digit+ ("." digit+)? (("E" | "e") ("+" | "-")? digit+)?

  space         += comment
  comment       = "//" (~"\n" any)*                 --single
                | "/*" (~"*/" any)* "*/"             --multi

  binaryOp    = "==" | "!=" | "<=" | "<" | ">=" | ">"
  addOp       = "+" | "-"
  mulOp       = "*" | "/" | "%"
}
```