Problem 31: Backus-Naur Form[1]

Source filename: `bnf.(cpp|java)`
Input filename: `bnf.in`
Output filename: `bnf.out`

Consider the following Backus-Naur Form (BNF) notation for the statements in an elementary programming language:

```
<statement>    :  <assignment> | HALT | PRINT(<variable>)
<assignment>   :  <variable> = <expression>
<expression>   :  <term> | <term><operator><expression>
<term>         :  <number> | <variable>
<variable>     :  x | y | z
<operator>     :  + | -
<number>       :  0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

(Note, the first entry reads, "A statement can be either an assignment, or the word `HALT`, or the word `PRINT` followed by a left parenthesis then a variable and then a right parenthesis. Any term in angular brackets, e.g. `<variable>`, will be defined somewhere in the BNF notation.

Write an interpreter for a source program whose statements are governed by the BNF above. Each statement of the source program will be on a separate line in the text file, `bnf.in`. Blanks within statements and blank lines are meaningless. The last statement in the input file will be `HALT`. The interpreter should create a source listing, `bnf.out`, of the program and to the right of each source statement the interpreter should display any output that the program would generate with a `PRINT` statment. If an error is found in a statement, the interpreter should display an error message to the right of the source statement. Two error messages are possible:

- `Bad syntax`
- `<variable> is undefined`

After handling any errors, the interpreter should resume interpretation with the next statement. Even if a statement contains multiple errors, display only one error message, giving priority to syntax errors. The source listing along with statement results and error messages should be displayed on the screen as well as placed in the output file, `bnf.out`.

Example Input File (`bnf.in`)

```
x = 4
PRINT(x)
z = x + y
PRINT(z)
z : y
y = x + x
z = x + y + y - y
PRINT(z)
HALT
```

Example Output File (`bnf.out`)

```
x = 4
PRINT(x)                    4
z = x + y                   y is undefined
PRINT(z)                    z is undefined
z : y                       Bad syntax
y = x + x
z = x + y + y - y
PRINT(z)                    12
HALT
```

[1] This problem appeared in an ACM South Central Regional Programming Contest.