A Simple Language for Compilation to the PC-231

Here is the abstract context-free term structure:

```
E  ::=  V  |  N  |  READ  |  WRITE E  |  V P  |  P V  |
        E O E  |  V = E  |  E ? E : E  |  ( E )
V  ::=  [a-z]+
N  ::=  [0-9]+
P  ::=  ++  |  --
O  ::=  +  |  -  |  *  |  <  |  >  |  ==  |  &  |  |
```

(Note that the final vertical bar here is in the object language you're implementing, not the meta-language of the BNF notation being used to describe the grammar.)

This structure constitutes a language of arithmetic and logical expressions with:

- variables (x, y, ..., or other sequences of lowercase letters);
- literal constants (357, 1, 0, ...; positive only);
- input and output (READ returns a value interactively from the user; WRITE outputs the expression and returns its value);
- prefix and postfix operators for increment and decrement (as in Java, the prefix ones change the variable's value before its returned, the postfix ones after);
- infix operators (arithmetic, relational and logical, with the usual meanings);
- variable assignment, but only with a variable on the left (this returns the value on the right as its result, just like in Java);
- and, finally, conditional expressions (again, as in Java; note that only one of the two branches should be evaluated.

The operators will ultimately have precedences assigned (in the usual way); we don't need these yet, however, as we aren't parsing expressions, but rather just representing them internally. (Thus the parentheses are not explicitly represented in the trees, either, and will only be used later.)

The "mixed semantics" of integers and booleans is like that in C (and in some ways, like Java); more specifically, when consuming a value for boolean purposes (i.e., for logical operators or for the conditional expression), a zero value is taken as false and any non-zero value as true. On the other hand, when producing a boolean value (i.e., as the result of a relational or logical operator) a false value is given as zero, but a true value is always given as one.