COL100 Assignment 5

Due date: 13 February 2022

Our task in the next few assignments is to build an interpreter for a simple Python program with a limited syntax, simulating the computer's execution. Let us start this week by interpreting an input program consisting of a sequence of statements, with each statement being of the form:

VARIABLE = EXPRESSION

EXPRESSION is one of:
UNARY_OPERATOR TERM
TERM BINARY_OPERATOR TERM

BINARY_OPERATOR is one of: $+, -, *, /, >, <, >=, <=, ==$, !=, and, or
UNARY_OPERATOR is one of: $-$, not

TERM is one of: VARIABLE, INTEGER_CONSTANT, True, False
VARIABLE is a sequence of one or more letters
INTEGER_CONSTANT is a sequence of one or more numeric characters ('0' to '9')

Read the input file one statement at a time and INTERPRET ("execute") the statement by maintaining all the variables and values encountered in the DATA list.

Here is a possible list of actions taken by the interpreter for the input program:

```
x = 1+3
y = 4
x = 5
```

- On reading "`x = 1+3`":
  - insert elements "1" and "3" into the DATA list.
  - Add 1+3 to get 4. Insert "4" into the DATA list. Let this be stored in list position i (that is, DATA[i] contains 4).
  - insert element ("x", i) in the DATA list. This is a way to implement "reference" (x refers to "4").
- On reading "`y = 4`":
  - search for 4 in the DATA list. Locate it in position i (DATA[i] already contains 4).

  - insert list element (“y”, i) into the DATA list.

- On reading “`x = 5`”:
  - search for 5 in DATA. Since it is not present, insert new list element “5”. Let this be in list position j (that is, DATA[j] contains 5).
  - search for x in DATA. Replace (“x”, i) by (“x”, j). x now refers to “5”.

When the program completes, print out the current values of all the variables and the list of GARBAGE integer objects (used in the program but not referred to any more by any variable).

Note the following:

- The above language supports two types of variables: integer and boolean.
- All variables are scalar (no aggregate types such as list).
- Remember to handle statements with variables on the right hand side (e.g., “`x = y + 2`”, “`x = y + z`” and “`x = y`”).
- If there is any error in the input, print the error and terminate the program.