Assignment 2: Solving equations

In this assignment you will use linked lists to recognize and solve equations.

Not all equations are easily solvable, therefore we restrict ourselves to linear equations, such as $3x + 4 = 2 - x$, $-8x + 1 = 0$ and $x = 3x + 1$. For further simplification we require that the equations do not contain parentheses, so e.g. $4(x + 1) = 3$ is excluded. Moreover, we only solve linear equations in one variable, not e.g. $x + y + 3 = x - 5$.

The assignment has two parts and both should be submitted on Themis. The deadline for part 2 is one week later.

Part 1: recognizing and classifying the equation

First you have to write a recognizer that should recognize equations.

Grammar

The equations to be recognized in this assignment are generated by the following grammar:

⟨*equation*⟩ ::= ⟨*expression*⟩ '`=`' ⟨*expression*⟩ .

⟨*expression*⟩ ::= `[` '`-`' `]` ⟨*term*⟩ `{` '`+`' ⟨*term*⟩ | '`-`' ⟨*term*⟩ `}` .

⟨*term*⟩ ::= ⟨*nat*⟩ | `[` ⟨*nat*⟩ `]` ⟨*identifier*⟩ `[` '`^`' ⟨*nat*⟩ `]` .

Here ⟨*nat*⟩ is a non-negative number as defined in Section 2.4.2 of the Lecture notes.

You may use the functions described in Section 2.4.4 (Scanning) and Section 2.4.5 (Recognition) of the Lecture notes. They are available on Themis.

Alert!
Here we only want a special kind of expressions, defined by the line for ⟨*expression*⟩ above. This differs from the expressions used in Section 2.4 of the Lecture notes and by the function `_accept_expression` in the file `recognizeExp.py`.

Tip
You should thus copy `recognizeExp.py` to `recognizeEq.py` and adapt it to the grammar above. Hint: You need to change function `_accept_expression` and `_accept_term`. The functions `_accept_number`, `acceptIdentifier` and `_accept_symbol` do *not* have to be changed.

Your Task

Your program should not solve all equations generated by this grammar. In this first part your program should check

- whether the input is an equation, i.e. a production of ⟨*equation*⟩;
- if so, whether it is an equation in one variable;
- if so, what its degree is.

The communication with your program should run via a dialogue, as with the evaluation of arithmetical expressions in Section 2.4.5 and Section 2.4.6 of the Lecture notes.

Input and Output Example

input	corresponding **output**

`3 + x - 3x = - x^2 - 3x^3`
`2x^1 - 4x^0 = 7`
`x + y = 0`
`2 longvariablename = 6`
 	
`x^2 + 5x - 7`
`2x = 3 = 4y`
`= 3x + 7`
`x^2 + x == 3`
`x (x + 3) = 7`
`!`
	`give an equation: 3 + x - 3x = - x^2 - 3x^3`
	`this is an equation in 1 variable of degree 3`
	
	`give an equation: 2x^1 - 4x^0 = 7`
	`this is an equation in 1 variable of degree 1`
	
	`give an equation: x + y = 0`
	`this is an equation, but not in 1 variable`
	
	`give an equation: 2 longvariablename = 6`
	`this is an equation in 1 variable of degree 1`
	
	`give an equation:`
	`this is not an equation`
	
	`give an equation: x^2 + 5x - 7`
	`this is not an equation`
	
	`give an equation: 2x = 3 = 4y`
	`this is not an equation`
	
	`give an equation: 3x + 7`
	`this is not an equation`
	
	`give an equation: x^2 + x == 3`
	`this is not an equation`
	
	`give an equation: x (x + 3) = 7`
	`this is not an equation`
	
	`give an equation: !`
	`good bye`

Themis

The following files will be inserted by Themis automatically — you do not need to re-upload them and you should not have any conflicting definitions in your own files:

- `recognizeEq_test.py`
- `scanner.py`

You should complete the code in the file:

- `recognizeEq.py`

Part 2: solving the equations

Now extend the program for Part 1 by giving the solution in the case that it is an equation of one variable of degree 1. Print the answer, rounded to 3 decimals. When the equation is not solvable (i.e. there are no solutions, or infinitely many), you print: `not solvable`.

Tip

When printing real numbers with e.g. 3 decimals, Python will print -0.000 for numbers between -0.0005 and 0. To prevent this, converts these values to 0 before printing.

Input and Output example

input	corresponding **output**

`x - 34 = - 15x + 14`
`4y + 1 = y`
`10000z = -1`
`x^2 = 4`
`5x - 7`
`2x = 3 = 4y`
`= 3x + 7`
`!`
	`give an equation: x - 34 = - 15x + 14`
	`this is an equation in 1 variable of degree 1`
	`solution: 3.000`
	
	`give an equation: 4y + 1 = y`
	`this is an equation in 1 variable of degree 1`
	`solution: -0.333`
	
	`give an equation: 10000z = -1`
	`this is an equation in 1 variable of degree 1`
	`solution: 0.000`
	
	`give an equation: x^2 = 4`
	`this is an equation in 1 variable of degree 2`
	
	`give an equation: 5x - 7`
	`this is not an equation`
	
	`give an equation: 2x = 3 = 4y`
	`this is not an equation`
	
	`give an equation: = 3x + 7`
	`this is not an equation`
	
	`give an equation: !`
	`good bye`

The following file will be inserted by Themis automatically — you do not need to re-upload it and you should not have any conflicting definitions in your own files:

- `scanner.py`

Extra (up to 2 bonus points)

- Extend your solver to equations of degree 2:

input	corresponding **output**
x^2 = 4 2 long ^ 2 - 4long^1 = 7 10 x^2 - 10 = -10 two^2 - 2 = 2 = y^2 + x^2 = 5 x 25y + 12 = y !	give an equation: x^2 = 4 this is an equation in 1 variable of degree 2 solution: -2.000 and 2.000 give an equation: 2 long ^ 2 - 4long^1 = 7 this is an equation in 1 variable of degree 2 solution: -1.121 and 3.121 give an equation: 10 x^2 - 10 = -10 this is an equation in 1 variable of degree 2 solution: 0.000 give an equation: two^2 - 2 = 2 = this is not an equation give an equation: y^2 + x^2 = 5 x this is an equation, but not in 1 variable give an equation: 25y + 12 = y this is an equation in 1 variable of degree 1 solution: -0.500 give an equation: ! good bye

- Implement a solver for pairs of linear equations with two variables:

input	corresponding **output**
3x + y = 4 6y = 4 - 4x 27=2apples+14bananas-11 - apples = 3 bananas - 11 x = 10 a + x = 100 7a + 7b -5a = 5a - 3a 23b - 5a = 22a - 27a !	give an equation: 3 x + y = 4 give an equation: 6 y = 4 - 4 x solution: x = 1.429 and y = -0.286 give an equation: 27 = 2 apples + 14 bananas - 11 give an equation: - apples = 3 bananas - 11 solution: apples = 5.000 and bananas = 2.000 give an equation: x = 10 give an equation: a + x = 100 solution: x = 10.000 and a = 90.000 give an equation: 7 a + 7 b - 5 a = 5 a - 3 a give an equation: 23 b - 5 a = 22 a - 27 a not solvable give an equation: ! good bye