Homework #2

Due: 2017-04-18 23:59

How to submit this homework

You will attach 3 files in an email with the subject line "Homework #2 - YOUR_NAME" (replace "YOUR_NAME" with your real name) sent to `huang.uclax@gmail.com`:

- File 1: the screen shot of Problem 1 (Spyder installation), e.g. hw2-1.png
- File 2: the python program file of Problem 2, e.g. hw2-2.py. I will execute the file "as is" against a different input file "hw2.txt" to verify your output.
- File 3: the answers to all questions of Problem 3 in a text file, e.g. hw2-3.txt.

Problem 1

Install spyder

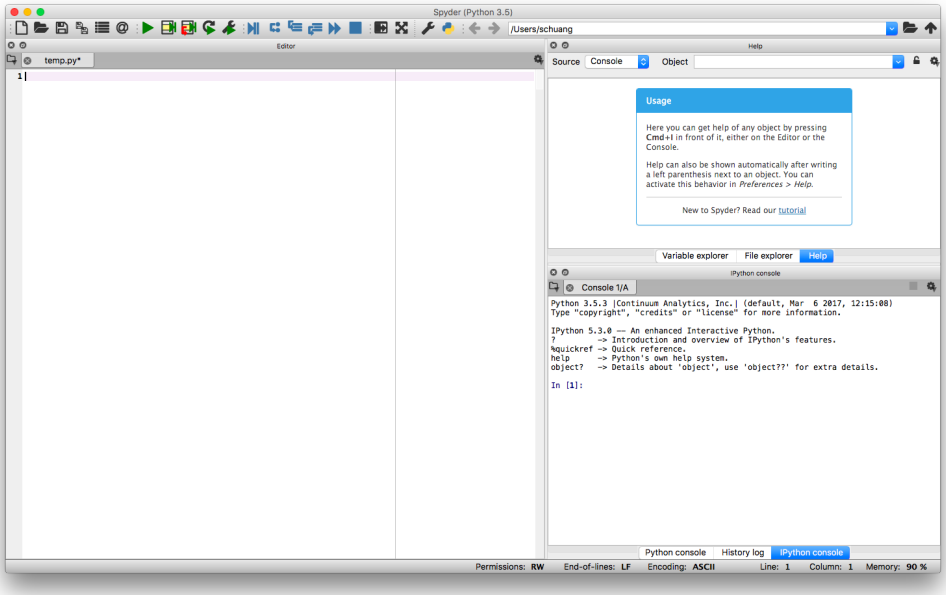
"spyder" is a (free) Python integrated development environment (IDE). Now that you already installed miniconda (in Homework #1), issue the following command to install spyder in a terminal:

```
conda install spyder
```

After the installation is completed, issue the command in the same terminal:

```
spyder
```

In a few seconds, the spyder window should appear that looks similar to:



Submit the screenshot image file with the rest of this homework.

Problem 2

The text file `hw2.txt` contains a number of Python expressions, one on each line. Write a program to evaluate and print out the values of all expressions. For example, suppose the file `hw2.txt` contains:

```
2 + 3 * 5
3**2 + 4 * 5
2 > 3
```

Your program's output would be:

```
17
29
False
```

Use following steps to construct and test your program:

1. Enter the following contents into a text file `hw2.txt`:

   ```
   2 + 3 * 5
   3**2 + 4 * 5
   2 > 3
   ```

   Note: the file must be a text file. To create the text file, you can use the Spyder IDE (see Problem 1), but not a word processor (e.g. Microsoft Word).

   Note: Do not contains any empty lines. Do not add an extra empty lines beyond the last expression.

2. Open the file using `open()` as a read-only file.
3. Read all file contents into a list using `readlines()`.
4. For each item in the list
   - remove possible leading and trailing white spaces by the string method `strip()`.
   - For each (stripped) item, evaluate the expression using the built-in function `eval()`.
   - Print the result on the screen.

Hint: your program may look something like:

```
... = open('hw2.txt', 'r')
... = f.readlines()
for r in ...:
    ... = r.strip()
    ... = eval(...)
    print(...)
```

Verify your program

Run your program with `hw2.txt`. The output should look like:

```
17
29
False
```

If you get the error message,

```
SyntaxError: unexpected EOF while parsing
```

you probably have empty lines in `hw2.txt`. Remove the empty lines in hw2.txt and try again.

Try changing the contents of `hw2.txt`. Re-run the program. Does the results reflect your changes? If everything looks correct, submit the Python program file with the rest of this homework.

Bonus points

What if there are empty lines in `hw2.txt`? How would you modify the code to ignore the empty lines and still produce correct output? For example, if `hw2.txt` contains an empty line between `2 + 3*5`: and `3**2 + 4*5`

```
2 + 3*5

3**2 + 4 * 5
2 > 3
```

Modify your program so it ignores the empty lines and generates the same output:

```
17
29
False
```

Hint: Running `strip()` on the empty line will result in a 0-length string (i.e. `len(s) == 0`).

Problem 3

For eacth set of Python commands, write down the final value of `x`. If the expression is invalid, explain why.

1. ```
   x = [1,2] + [3,4]
   ```
2. ```
   x = [hi]*3
   ```
3. ```
   x = [3,5,7]
   x.append(8)
   ```
4. ```
   x = {'x': 3, 'y':5}
   x['p'] = 7
   ```
5. ```
   x = 'hello'
   x[1] = 'a'
   ```
6. ```
   x = 2 > 3 and (4 < 5 or 6 > 7)
   ```
7. ```
   y = 'alasfjsdlafjajoiwjrofm;flskajfsakdjckljfd'
   x = y[-2]
   ```
8. ```
   x = 'hello' + 'world'
   ```
9. ```
   y = [1, 2, 3, [5, 6, 7]]
   x = y[3][-1]
   ```
10. ```
    y = [1, 2, 3, [5, 6, 7]]
    x = y[3][-1]
    ```
11. (Using Python 3)
    ```
    x = 5 / 3
    ```
12. ```
    y = (3,5,7)
    x = y[0]
    ```
13. ```
    x = 3 ** 2 * 5  / 10 + 5
    ```
14. ```
    a = 5
    b = 2.5
    c = a + b
    x = type(c)
    ```
15. ```
    a = [1,2,3]
    x = a[3]
    ```

16.
```
a = ['cat', 'dog', 'fox']
x = 'CAT' in a
```

17.
```
a = ['cat', 'dog', 'fox']
x = ['cat', 'dog'] in a
```

18.
```
a = [1,2,3,4,5]
b = a[2:]
x = len(b)
```