Lab08: Loops, Strings and Lists

Prelab 8

We encourage you to work together on the Pre Lab. The Pre Lab is not graded but will help you prepare for your lab session. If you have any questions on the material of the Pre lab, first check the book and recitation slides. If you continue to have any doubt about it, please email your recitation TA or the course instructor.

This prelab explains while loop and how to access (that is how to read and/or write) elements in a list of lists.

Here is PreLab8

Environment Setup

Go to your working directory in “data.cs.purdue.edu” and create a directory “cs177/lab08”. Refer to the first lab material if you need to remember the steps to do so. Then start the IDLE Python Interpreter.

Task 1: Print a matrix with a special order

In this exercise, you will write a function called print_matrix() that takes a matrix as an input parameter, and prints the matrix with a special order. The input matrix is always a list of lists containing integers, an example matrix is as follows.

```
matrix = [[1, 2, 3, 4],
          [5, 6, 7, 8],
          [9, 10, 11, 12],
          [13, 14, 15, 16]]
```

The order should be followed is shown in the following figure.
Following the above logic, the output of the given matrix is:

<table>
<thead>
<tr>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
</tr>
<tr>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

Make sure your program works for different matrices as well. Another example input/output as follows:

```python
matrix = [[80, 70, 46, 98, 35],
          [63, 76, 14, 99, 72],
          [4,  48, 19, 32, 51],
          [48, 48, 82, 80, 49],
          [34, 96, 33, 17, 30],
          [49, 59, 16, 6,  64]]
```

Output:

```
80
70
46
98
35
72
51
49
30
64
6
16
59
49
```
Please notice that there is no repetition for the elements in the corners at the output.

More specifically, you need to write the following function,

```python
def print_matrix(matrix):
    # TODO
    # TODO
```

After you finish implementing the `print_matrix()` function, write a `main()` function that calls the `print_matrix()` function with the above matrix and make sure it produces the same output.

Save the program in `cs177/lab08/print_matrix.py`

**Task 2: Count number of smaller**

Consider we have two lists, X and Y, containing integer values. We want to calculate Z, which contains for each element x in X, number of elements in Y smaller than x. Some examples are as follows:

**Example 1:**

Input:

```
X = [5, 20, 14, 1, 7]
Y = [2, 12, 9, 5, 4, 16]
```

Output:

```
Z = [2, 6, 5, 0, 3]
```

As seen in Example 1, there are 2 numbers in Y which is smaller than 5. And similarly, there are 6 numbers in Y which are smaller than 20.

**Example 2:**

Input:

```
X = [3, 15, 2, 9, 19]
Y = [12, 7, 1, 23, 10]
```

Output:

```
Z = [1, 4, 1, 2, 4]
```

**Example 3:**

Input:

```
X = [16, 2, 8, 13, 5, 22, 29]
Y = [5, 13, 6, 3, 5, 25]
```
Output:
\[ Z = [5, 0, 4, 4, 1, 5, 6] \]

More specifically, you need to write the following function,

```python
def count_number_of_smaller(X,Y):
    # TODO
    # TODO
    return Z
```

After you finish implementing the `count_number_of_smaller()` function, write a `main()` function that calls the `count_number_of_smaller()` function with one of the above input and make sure it produces the same output.

Save the program in `cs177/lab08/count_number_of_smaller.py`

**Task 3: Substitution as a list of pairs**

Previously, you have learned the replace string method that substitutes the new letters with the old matching ones, for example:

```python
>>> myStr = "Python Program"
>>> myStr = myStr.replace('o', '*')
>>> print(myStr)
Pyth*n Pr*gram
```

One limitation of replace is: it can only replace one substring a time. What if we want to replace `o` with `*` and `P` with `$`?

In this exercise, you will write a function called `substitute_pairs()` that receives a string and list of pairs as a parameter as follows:

```python
pairs = [['P', '$'], ['o', '*']]
```

which means, each `P` in the input string will be substituted with a `$` and each `o` will be substituted with a `*`. To make things simple, we substitute one letter. Note that you cannot use any string methods (e.g., find, replace, etc.) in this exercise, that means you can only use for loop, and other control structures to solve this problem. Here’s a complete example:

```python
pairs = [['P', '$'], ['o', '*']]
my_str = "Python Program"
my_str = substitute_pairs(my_str, pairs)
print(my_str)
```

The output of the above Python program should be:

```
$yth*n $r*gram
```
More specifically, you need to write the following function,

```python
def substitute_pairs(my_str, pairs):
    # TODO
    # TODO
    return result
```

After you finish implementing the `substitute_pairs()` function, write a `main()` function that calls with the `substitute_pairs()` function with the above inputs and make sure it produces the same output.

Save the program in `cs177/lab08/substitute_pairs.py`

**Turnin Instructions**

Run putty and login to data.cs.purdue.edu. Turn in your lab by typing:

```
$ cd cs177
$ turnin -c cs177=COMMON -p lab08 lab08
```

To verify that you did not submit a wrong file or an empty one, run the following command:

```
$ turnin -v -p lab08
```

**Grading Criteria**

<table>
<thead>
<tr>
<th>Task</th>
<th>Points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task 1</td>
<td>30</td>
</tr>
<tr>
<td>Task 2</td>
<td>30</td>
</tr>
<tr>
<td>Task 3</td>
<td>40</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
</tr>
</tbody>
</table>

From: [Computer Science Courses](http://courses.cs.purdue.edu/)

Permanent link: [http://courses.cs.purdue.edu/cs17700:fall15:labs:lab08](http://courses.cs.purdue.edu/cs17700:fall15:labs:lab08)

Last update: 2015/10/13 17:04