This problem will give you practice with indexing and slicing. Begin by downloading this file: `ps1pr2.py`. Open it in IDLE, as discussed in class.

List puzzles

The first half of the problem focuses on lists. In particular, you will be constructing new lists from the following:

```
pi = [3, 1, 4, 1, 5, 9]
e = [2, 7, 1]
```

We've given you these lines in `ps1pr2.py`. In addition, we've provided the answer to the first puzzle (puzzle 0). You should add in the answers for the remaining puzzles, *following the format that we've given you for puzzle 0*.

*The expressions that you construct must **only** use* `pi` *and* `e` *and the following list operations:*

- list indexing (e.g., `pi[0]`)
- list slicing (e.g., `e[1:]` or `pi[2:4]`)
- skip-slicing (e.g., `pi[6:4:-1]`)
- list concatenation (e.g., `pi[:1] + e[1:]`. Note that you may *not* use + to add values numerically.)
- list construction (e.g., `[e[2]]` or `[e[2], e[0]]`)

We encourage you to try using as few operations as possible, to keep your answers elegant and efficient. However, you will get full credit for *any* expression that follows the rules above and produces the correct result.

Before getting started, you should run `ps1pr2.py` in IDLE using *F5*. This will make the lists `pi` and `e` available to you in the Python Shell.

Here are the puzzles:

1. Use `pi` and/or `e` to create the list `[2, 5, 9]`, and assign it to the variable `answer0`. We've given you the code for this puzzle.

2. Use `pi` and/or `e` to create the list `[2, 7]` and assign it to the variable `answer1`. Your answer should follow the format that we've given you for problem 0. In other words, it should look like this:

   ```
   # Puzzle 1:
   # Creating the list [2, 7] from pi and e
   answer1 =
   ```

```
print(answer1)
```

where you put the appropriate expression to the right of the assignment operator (=). **Please leave a blank line between puzzles to make things more readable.**

3. Use `pi` and/or `e` to create the list `[5, 4, 3]`, and assign it to the variable `answer2`. Here again, make sure to follow the correct format, and to leave a blank line between puzzles.

4. Use `pi` and/or `e` to create the list `[3, 5, 7]`, and assign it to the variable `answer3`. (*Optional challenge:* See if you can do this with just three list operations!)

5. Use `pi` and/or `e` to create the list `[1, 2, 3, 4, 5]`, and assign it to the variable `answer4`. (*Optional challenge:* See if you can do this with just three list operations!)

String puzzles

The second half of the problem focuses on strings. In particular, you will be working with the following strings:

```
b = 'boston'
u = 'university'
t = 'terriers'
```

We've given you these lines in `ps1pr2.py`, along with the answer to the first string puzzle (puzzle 5). Run the file as needed so that the strings will be available for you to experiment with in IDLE.

*The expressions that you construct for the remaining puzzles must **only** use the above strings and the following string operations:*

- string indexing (e.g., `b[0]`)
- string slicing (e.g., `u[1:]` or `t[2:4]`)
- skip-slicing (e.g., `u[6:4:-1]`)
- string concatenation (e.g., `b + u`)
- string repetition (e.g., `5*b` or `3*u[-1]`)

Here again, you will get full credit for *any* expression that follows the rules above and produces the correct result, but we encourage you to try using as few operations as possible.

Here are the puzzles:

1. Use b, u, and/or `t` to create the string `'bossy'`, and assign it to the variable `answer5`. We've given you the following code for this puzzle:

   ```
   # Puzzle 5:
   # Creating the string 'bossy'
   answer5 = b[:3] + t[-1] + u[-1]
   print(answer5)
   ```

   Note that our answer involves 5 operations: 2 uses of indexing, 1 slice, and 2 concatenations with +. (It's actually possible to solve this puzzle using only 3 operations. Give it a try if you have time!)

2. Use b, u, and/or `t` to create the string `'universe'`, and assign it to the variable `answer6`. (Our best answer uses 3 ops.) Here again, make sure to follow the correct format, and to leave a blank line between puzzles.
3. Use b, u, and/or `t` to create the string `'roster'`, and assign it to the variable `answer7`. (Our best: 5 ops.)
4. Use b, u, and/or `t` to create the string `'boisterous'`, and assign it to the variable `answer8`. (Our best: 8 ops.)
5. Use b, u, and/or `t` to create the string `'yesyesyes'`, and assign it to the variable `answer9`. (Our best: 4 ops.)
6. Use b, u, and/or `t` to create the string `'trist'`, and assign it to the variable `answer10`. (Our best: 4 ops.)

After finishing all of the puzzles, make sure to run your `ps1pr2.py` file to check that the correct outputs are printed.