Problem 1: Sequences and Data

This part of the homework will give you experience splicing and interacting with Python data.

When you're finished with this assignment, submit your code at the bottom of this page.

Slicing and Indexing Challenges

This problem will exercise your slicing-and-indexing skills.

First, create a copy of this trinket.

Please do include your name at the top of the file.

After the initial comment, this code defines the list named `pi` and the list named `e`.

When you run the file, the line `answer0 = [ e[0] ] + pi[-2:]` (line 14) will define the value held by the variable `answer0`.

Then, the code will print the value of the variable `answer0`.

The problems below ask you to create several lists using **only** the list named `pi`, the list named `e`, and these list operations:

- list indexing such as `pi[0]`
- list slicing such as `e[1:]`
- skip-slicing such as `pi[6:4:-1]`
- list concatenation, `+`, such as `pi[1:] + e[1:]` (Do not use `+` to add values numerically.)
- the list-making operator, `]` , for example: `[ , ]` `[ e[2], e[0] ]`

For each problem, place your answer into an appropriate variable in the same style as the example answer (`answer0`), including comments. Use copy-and-paste to be as efficient as possible!

Please **leave a blank line** or two between your answers (to keep things readable)!

Remember to **run your file** to check your answers often and to try things out.

For fun only, you might try using as few operations as possible, to keep your answers elegant and efficient. This is **not required**.

The List Problems

0. Use `pi` and/or `e` to create the list `[2,5,9]`. This is the example above, stored in the variable `answer0`.
1. Use `pi` and/or `e` to create the list `[7,1]`. Store this list in the variable `answer1`. Again, this means that you should have these three lines in your file—but with the answer included in the appropriate spot, after the `answer1 =`. Copy-and-paste from here, if you'd like:

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

2. Use `pi` and/or `e` to create the list `[9,1,1]`. Store this list in the variable `answer2`.
3. Use `pi` and/or `e` to create the list `[1,4,1,5,9]`. Store this list in the variable `answer3`.
4. Use `pi` and/or `e` to create the list `[1,2,3,4,5]`. Store this list in the variable `answer4`.

You must assign your correct solution to the given variable using an equals sign (answer 1 = ?) or else your assignment will not be graded as correct.

Practicing with Strings

This problem continues in the style of the last one, but uses strings rather than lists. First, copy these strings into your trinket underneath the previous problems (with some blank lines to keep things apart!):

```
# starting strings for Homework 1

h = 'harvey'
m = 'mudd'
c = 'college'
```

You may use any combination of these four string operations:

- String indexing, e.g., `h[0]`
- String slicing, e.g., `m[1:]`
- String concatenation, `+`, e.g., `h + m`
- Repetition, `*`, e.g., `42*c` (using integers is okay here)

Again, less is more. The number of operations in the shortest answers that we know about are in parentheses in the string problems below. If you'd like, you might see if your answers are equally or more concise, but this is optional. **Any correct answer is okay**—there's no requirement to use a small number of operations.

Example Problem

Use `h`, `m`, and `c` to create `'heyyou'`. Store this string in the variable `answer5`. We used 9 operations.

Answer

Please copy and paste this into your trinket:

```
# Problem 5:
# Creating the string 'heyyou'
answer5 = h[0] + h[4:] + h[-1] + c[1] +
m[1]
print answer5
```

The 9 operations are 4 uses of list indexing, 1 slice, and 4 concatenations with `+`.

The String Problems

Here are the string-creation challenges (and, in parens, our most efficient answers, at least so far):

Remember that the "most efficient answers" are not at all needed (they may be fun, but any working answer is 100% okay!)

5. (The example from above) Create `'heyyou'` and store this string in the variable `answer5`. (our best: 7 ops.)
6. Create `'collude'` and store this string in the variable `answer6`. (our best: 5 ops.)
7. Create `'arveyudd'` and store this string in the variable `answer7`. (our best: 3 ops.)
8. Create `'hardeharharhar'` and store this string in the variable `answer8`. (our best: 7 ops.)
9. Create `'legomyego'` and store this string in the variable `answer9`. (our best: 8 ops.)
10. Create `'clearcall'` and store this string in the variable `answer10`. (our best: 8 ops.)

Submit Homework 2, Problem 1

20.0/20.0 points (graded)

To submit Homework 2, Problem 1, you'll need to copy your code from your trinket and paste it into the box below. After you've pasted your code below, click the "Check" button.

IMPORTANT: Make sure that there aren't spaces at the beginning of your code, and that you copied all of the characters. If there are extra spaces or you are missing spaces, our server won't be able to run your code and we won't be able to give you any of the points you deserve for your hard work.

1

2

3

4

```
5
6
7
8
9
pi =
[3,1,4,1,5,9]
10
e =
[2,7,1]
11
12
13
14
answer0 = [ e[0] ] + pi[-2:]
15
print answer0
```

16

17

18

19

```
answer1 = e[1:]
```

20

```
print answer1
```

21

22

23

24

```
answer2 = pi[-1:0:-2]
```

25

```
print answer2
```

Press ESC then TAB or click outside of the code editor to exit
correct

correct

Test results
CORRECT See full outputSee full output

You have used 1 of 3 attempts Some problems have options such as save, reset, hints, or show answer. These options follow the Submit button.