Extracted from:

Python Brain Teasers

Exercise Your Mind

The Pragmatic Bookshelf

Raleigh, North Carolina

The Pragmatic Programmers

Python Brain Teasers

Exercise Your Mind

Miki Tebeka

edited by Margaret Eldridge

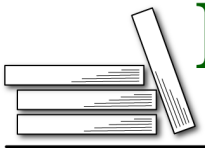
Python Brain Teasers

Exercise Your Mind

Miki Tebeka

The Pragmatic Bookshelf

Raleigh, North Carolina



Pragmatic Bookshelf

For our complete catalog of hands-on, practical, and Pragmatic content for software developers, please visit *https://pragprog.com*.

The team that produced this book includes:

CEO: Dave Rankin
COO: Janet Furlow
Managing Editor: Tammy Coron
Development Editor: Margaret Eldridge
Copy Editor: Jennifer Whipple
Indexing: Potomac Indexing, LLC
Layout: Gilson Graphics
Founders: Andy Hunt and Dave Thomas

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

ISBN-13: 978-1-68050-900-7
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—September 2021

Puzzle 4

A Task to Do

tasks.py

```
Line 1  from heapq import heappush, heappop
-
-       tasks = []
-       heappush(tasks, (30, 'work out'))
5       heappush(tasks, (10, 'wake up'))
-       heappush(tasks, (20, 0xCAFFE))
-       heappush(tasks, (20, 'feed cat'))
-       heappush(tasks, (40, 'write book'))
-
10      while tasks:
-           _, payload = heappop(tasks)
-           print(payload)
```

Guess the Output

Try to guess what the output is before moving to the next page.

This code will raise a TypeError exception.

The built-in heapq module implements min-heap over lists.

It's common to use a heap for a priority queue. Pushing and deleting from the heap are log(N) operations, and the first item in the heap (e.g., tasks[0]) is always the smallest.

To compare items in the heap, heapq uses the comparison defined in the object's type (using the < operator, which maps to the specific type's __lt__ special method). The objects in the tasks heap are tuples. Python orders tuples, and lists, in a lexicographical order, very much like books are ordered in the library. Lexicographical order compares the first two items, then the second two, and so on. Finally, if all of the items are equal, the longer tuple is considered bigger.

In line 11, you pop the first item from tasks, which is (10, 'wake up'). After this item is removed from the heap, heapq will move the smallest item to the top of the heap. There are two candidates (20, 'feed cat') and (20, 0xCAFFE); since the first items in these tuples are equal, Python will try to compare the second items.

l33t Code

0xCAFFE is a hexadecimal (base 16) number. Writing "English" this way is called "leet" (or "l33t").

Comparing 'feed cat' (a str) with 0xCAFFE (an int) will raise an exception.

Further Reading

heapq Module
docs.python.org/3/library/heapq.html

Heap Data Structure on Wikipedia
en.wikipedia.org/wiki/Heap_(data_structure)

Lexicographical Order on Wikipedia
en.wikipedia.org/wiki/Lexicographical_order

Tuples and Sequences
docs.python.org/3/tutorial/datastructures.html#tuples-and-sequences