```
treasures.put("cove", 75.);
treasures.put("x", 100.);
treasures.put("palm", treasures.get("palm") + treasures.size());
treasures.remove("beach");
double found = 0;
for (double treasure : treasures.values()) {
    found += treasure;
}
```

What is the value of...

treasures.get(x)

54

✓ **Answer:** 54

54

treasures.get("x")

100.

✓ **Answer:** 100

100.

found

229

✓ **Answer:** 229

229

Explanation

After the first four `put()` calls, the map has stored the pairs ("beach", 25), ("palm", 50), ("cove", 75), ("x", 100). The fifth `put()` call adds the size of the map (4) to the entry for `"palm"`, so that entry is now ("palm", 54). Finally the entry for "beach" is removed from the map, so the final state of the map is ("palm", 54), ("cove", 75), ("x", 100).

Now that we know what the map looks like, we can answer the questions. `treasures.get(x)` returns the value stored for the key `"palm"`, which is 54. `treasures.get("x")` returns the value stored for `"x"`, which is 100. Finally, `found` sums up all the values currently stored in the map, which is 54+75+100 = 229. You can see this code in action in Online Java Tutor.

Submit