MIT Open Learning Library

My Courses   All Courses   About   Contact   ♥ Give   Help   **vitorpbarbosa7**

Course   Progress

Course > Readings/Videos > Reading 3: Testing > Questions

‹ Previous   🎞   🎞   ✏   🎞   ✏   🎞   ✏   🎞   ✏   🎞   ✏   ⊞   ✏   🗐   Next ›

Questions

🔖 Bookmark this page

partitioning

1/1 point (graded)

Consider the following specification:

```
/**
 * Reverses the end of a string.
 *
 *                       012345                    012345
 * For example: reverseEnd("Hello, world", 5) returns "Hellodlrow ,"
 *                              <----->                    <----->
 *
 * With start == 0, reverses the entire text.
 * With start == text.length(), reverses nothing.
 *
 * @param text    non-null String that will have its end reversed
 * @param start   the index at which the remainder of the input is reversed,
 *                requires 0 <= start <= text.length()
 * @return input text with the substring from start to the end of the string reversed
 */
public static String reverseEnd(String text, int start)
```

Which of the following are reasonable partitions for the start parameter? Check all that apply.

- [ ] start = 0, start = 5, start = 100

- [ ] start < 0, start = 0, start > 0

- [x] start = 0, 0 < start < text.length(), start = text.length()

- [ ] start < text.length(), start = text.length(), start > text.length()

✔

Explanation
0, 5, 100 is not a partition. A partition should be a division of the whole space of possible start values, not specific test cases.
start < 0 and start > text.length() are not legal inputs for the function, so it isn't reasonable to use them as part of a partition.

[Submit]   ⓘ Show Answer

ⓘ Answers are displayed within the problem

partitioning, part 2

1/1 point (graded)

Which of the following are reasonable partitions for the text parameter? Check all that apply.

- [ ] text contains some letters; text contains no letters, but some numbers; text contains neither letters nor numbers

- [x] text.length() = 0; text.length() > 0

- [x] text.length() = 0; text.length()-start is odd; text.length()-start is even

- [ ] text is every possible string from length 0 to 100

✔

Explanation
Letters and numbers aren't important to the behavior of this function, so it isn't reasonable to partition on that property.
Length is a useful partition, since it can interact with the start parameter.
Partitioning on even and odd length is also reasonable, because reversing an odd-length substring has different behavior (since it leaves the middle element in place) than an even-length string (where all elements swap).
Partitioning into all possible length 0 to 100 strings will produce far too many test cases.

Submit

Show Answer

❶ Answers are displayed within the problem

‹ Previous   **Next ›**

👤 🕐   Some Rights Reserved

Open Learning Library

About

Accessibility

All Courses

Why Support MIT Open Learning?

Help

Connect

Contact

Twitter

Facebook