

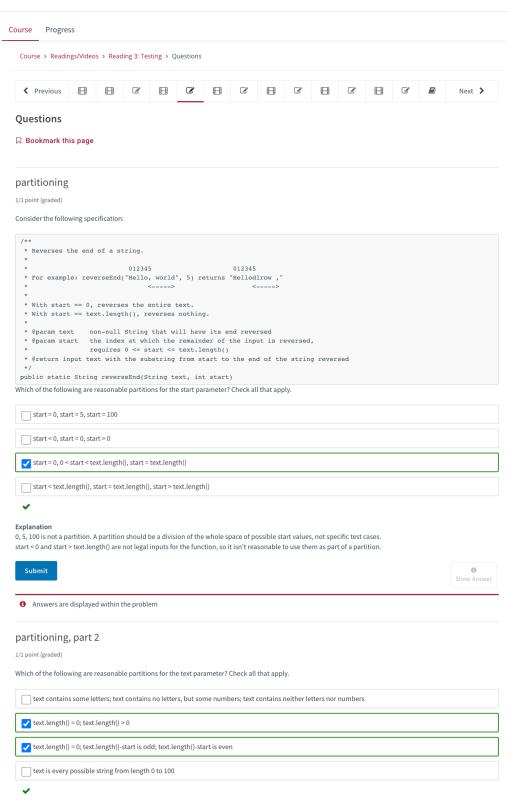
A (S) Some Rights Reserved

Open Learning Library	Connect
About	Contact
Accessibility	Twitter
All Courses	Facebook
Why Support MIT Open Learning?	
Help	
Privacy Policy Terms of Service	

Open Learning Library

Help vitorpbarbosa7

My Courses All Courses About Contact Give



# 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

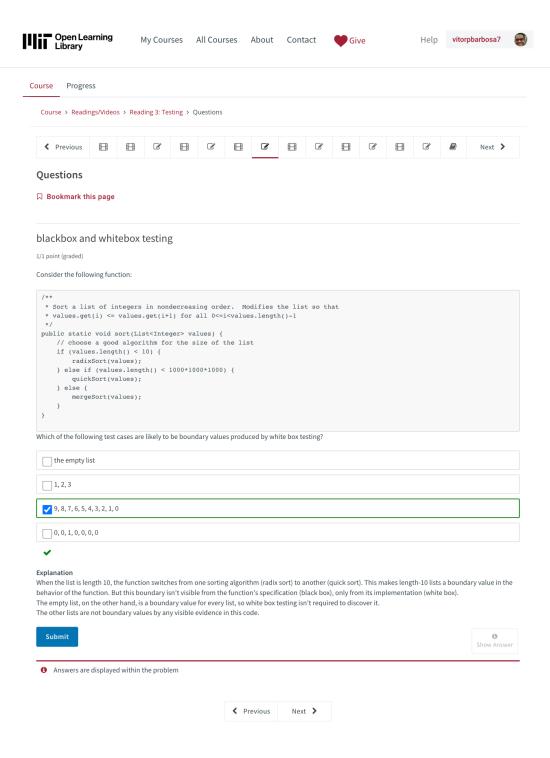
Partitioning on even and odd length is also reasonable, because reversing an odd-length substring has different behavior (since it leaves the middle element in place) than an even-length string (where all elements swap).

Partitioning into all possible length 0 to 100 strings will produce far too many test cases.



△ ③ Some Rights Reserved

Open Learning Library Connect About Contact Accessibility Twitter All Courses Facebook Why Support MIT Open Learning? Help



△ ○ Some Rights Reserved

Open Learning Library
About
Contact
Accessibility
Twitter
All Courses
Why Support MIT Open Learning?

Page 2
Questions | Reading 3: Testing | 6.005.1x Courseware | MIT Open Learning Library
https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.005.1x+3T2016/courseware/Readings\_Videos/03-Testing/?activate\_block\_id=block-v1%3AMITx%2B6.005.1x%2B3T2016%2Btype%40sequential%2Bblock...

Help



My Courses All Courses About Contact



Help vitorpbarbosa7





## Questions

### □ Bookmark this page

## coverage

3/3 points (graded)

#### Install EclEmma in Eclipse.

Then create a new Java class called Hailstone.java containing this code:

```
public class Hailstone {
  public static void main(String[] args) {
  int n = 3;
  while (n != 1) {
          if (n % 2 == 0) {
    n = n / 2;
            n = n / 2,
} else {
n = 3 * n + 1;
```

Run this class with EclEmma code coverage highlighting turned on, by choosing Run / Coverage As / Java Application.

By changing the initial value of n, you can observe how EclEmma highlights different lines of code differently.

When n=3 initially, what color is the line n = n/2 after execution?



## Explanation

If it's hard to perceive the difference between the red and green highlighting, you can change the color of the coverage highlighting by going to Preferences / General / Appearance / Editors / Text Editors / Annotations. The annotations to change are called Full Coverage, Partial Coverage, and No Coverage. You might, for example, make Full Coverage white, Partial Coverage light gray, and No Coverage dark gray.

When n=16 initially, what color is the line n = 3 \* n + 1 after execution?



## Explanation

Run EclEmma and find out!

What initial value of n would make the line while (n != 1) yellow after execution?



Yellow is used for a line that contains a branch (e.g., an if or while predicate) that is only taken in one direction during the program -- either the predicate is false every time the program gets there, or the predicate is true every time. In this case, the way to make the while predicate yellow is to make n != 1 false, so that the predicate executes just once and the loop is skipped entirely. n=1 does that.



Next >

Previous

Page 2
Questions | Reading 3: Testing | 6.005.1x Courseware | MIT Open Learning Library
https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.005.1x+3T2016/courseware/Readings\_Videos/03-Testing/?activate\_block\_id=block-v1%3AMITx%2B6.005.1x%2B3T2016%2Btype%40sequential%2Bblock.

△ ③ Some Rights Reserved

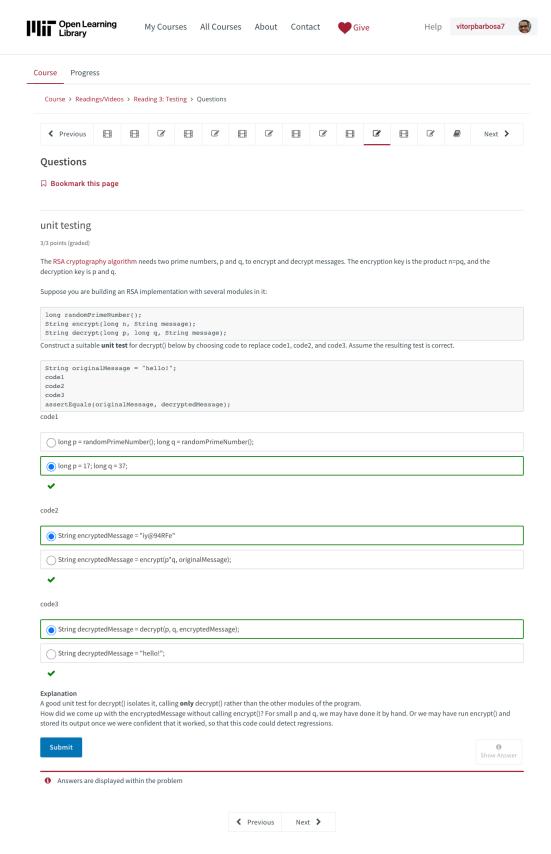
Open Learning Library About Contact Accessibility Twitter All Courses Facebook Why Support MIT Open Learning? Help

Previous											Ø				Next
uestions															
Bookmark th	nis page														
it testing															
points (graded)															
RSA cryptogra cryption key is		rithm nee	eds two p	orime num	bers, p a	nd q, to e	encrypt ar	nd decryp	ot messag	ges. The e	ncryptio	n key is th	ie produc	ct n=pq, a	nd the
pose you are	building a	n RSA imլ	olementa	ation with	several n	nodules i	in it:								
ong randomP															
tring encry	pt(long	p, long	q, St	ring mes											
nstruct a suital	ole <b>unit te</b>	est for dec	rypt() be	elow by ch	oosing co	ode to re	place cod	le1, code2	2, and coo	de3. Assu	me the re	esulting te	est is corr	ect.	
tring origi: ode1 ode2 ode3	nalMessa	age = "h	ello!"	;											
ssertEquals e1	(origina	alMessag	ge, dec	ryptedMe	essage);										
61															
long p = ran	ıdomPrim	eNumber	·(); long c	q = randon	nPrimeNu	umber();									
			·(); long c	q = randon	nPrimeNu	umber();									
			·(); long c	q = randon	nPrimeNu	umber();									
long p = 17;			(); long c	q = randon	nPrimeNu	umber();									
long p = 17;	long q = 3	37;			nPrimeNu	umber();									
long p = 17;  le2  String encry	long q = 3	sage = "iy	@94RFe'												
long p = 17;  de2  String encry	long q = 3	sage = "iy	@94RFe'												
long p = 17;  de2  String encry	long q = 3	sage = "iy	@94RFe'												
long p = 17;  String encry  String encry	long q = 3	sage = "iy	@94RFe'												
long p = 17;  le2  String encry  String encry	long q = 3	sage = "iye	@94RFe'	q, original!	Message)	•									
long p = 17;  le2  String encry  String encry	long q = 3	sage = "iye sage = dee	@94RFe'	q, original!	Message)	•									
long p = 17;  le2  String encry  String decry	long q = 3	sage = "iye sage = dee	@94RFe'	q, original!	Message)	•									
long p = 17;  le2  String encry  String decry  String decry	long q = 3	sage = "iye sage = dee	@94RFe'	q, original!	Message)	•									
long p = 17;  le2  String encry  String decry  String decry	long q = 3  yptedMess  yptedMess  yptedMess  yptedMess  yptedMess  yptedMess	sage = "iye sage = end sage = the sage = "he	@94RFe'	q, originall q, encrypte age withous	edMessage)	ge);	n the othe	nall p and	d q, we ma		one it by	hand. Or	we may	have run	encrypt() a
e2  String encry  e3  String decry  String decry  lanation  od unit test for did we come and its output of the come are dits output of the come are displayed as a second of the come are displayed as a	long q = 3  yptedMess  yptedMess  yptedMess  yptedMess  yptedMess  yptedMess	sage = "iye sage = end sage = the sage = "he	@94RFe'	q, originall q, encrypte age withous	edMessage)	ge);	n the othe	nall p and	d q, we ma		one it by	hand. Or	we may	have run	encrypt() a
long p = 17;  le2  String encry  String decry  String decry  String decry  I anation  ood unit test for which we come	or decrypted Messey pted Messe	sage = "iye sage = dec	@94RFe'	q, originall q, encrypte sage without t it worked	edMessage)	ge);	n the othe	nall p and	d q, we ma		one it by	hand. Or	we may	have run	•

△ ⑤ Some Rights Reserved

Open Learning Library Connect Contact About Accessibility Twitter Facebook All Courses Why Support MIT Open Learning?

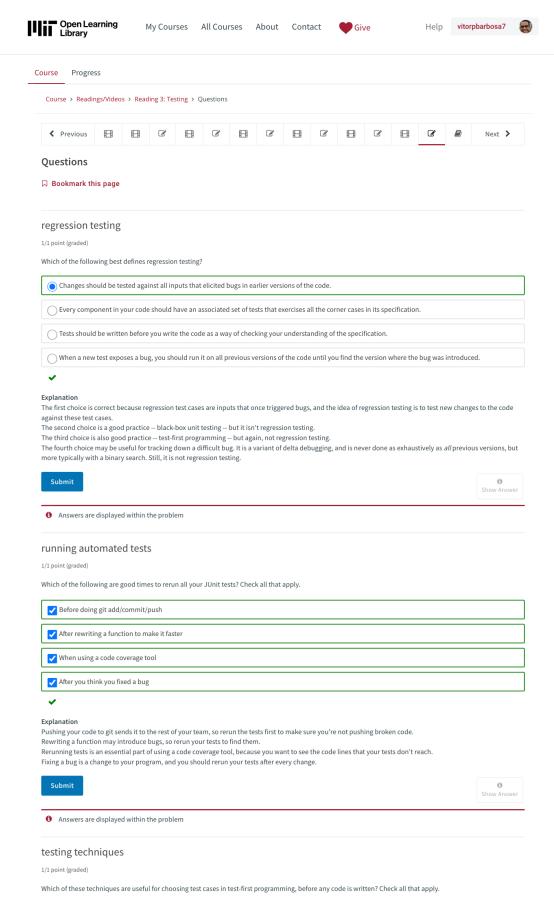
Help

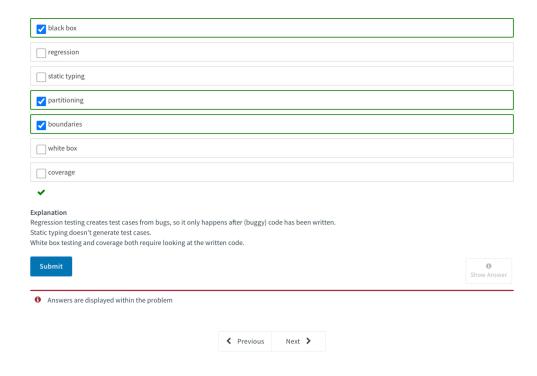


Page 2
Questions | Reading 3: Testing | 6.005.1x Courseware | MIT Open Learning Library
https://openlearninglibrary.mit.edu/courses/course-v1:MITx+6.005.1x+3T2016/courseware/Readings\_Videos/03-Testing/?activate\_block\_id=block-v1%3AMITx%2B6.005.1x%2B3T2016%2Btype%40sequential%2Bblock.

△ ③ Some Rights Reserved

Open Learning Library About Contact Accessibility Twitter All Courses Facebook Why Support MIT Open Learning? Help





 $\stackrel{\wedge}{\sim} \ \, \mathbb{O} \quad \, \text{Some Rights Reserved}$ 

