

## Lab 07. CS1110 Fall 2011. Practice with assertions and loops

### Goal of this lab

Practice with assertions and with loops that process a range of integers. Part of this lab is to be done on paper.

#### 1. Question on ranges:

How many values are in the following ranges? The last one requires a formula in terms of  $h$  and  $k$ . Remember: in the notation  $h..k$ , we require  $k \geq h-1$ . For example,  $5..4$  is OK but  $5..3$  is not allowed.

- (a)  $5..6$
- (b)  $5..5$
- (c)  $5..4$
- (d)  $h..h$
- (e)  $h..h-1$
- (f)  $h..k$

2. Each line below asks you to write an assignment statement. We did the first one for you.

Range	Want	Assignment statement
(a) $h..k$	Assign to $k$ so that the range has 1 element:	$k = h;$
(b) $h..k$	Assign to $h$ so that the range has 1 element:	
(c) $h..k$	Assign to $k$ so that the range has 0 elements:	
(d) $h..k$	Assign to $h$ so that the range has 0 elements:	
(e) $0..n-1$	Assign to $n$ so that the range has 1 element:	
(f) $0..n-1$	Assign to $n$ so that the range has 0 elements:	
(g) $h-1..10$	Assign to $h$ so that the range has 1 element:	
(h) $h+1..10$	Assign to $h$ so that the range has 0 elements:	

3. Each line below contains an assertion  $P$  that is guaranteed to be true. Each line also contains an assertion  $R$ , which we would like to be true. In the righthand column, put a Boolean expression that, when true, allows us to conclude that  $R$  is true. Recall that the notation  $s[k..]$  means the substring from character  $k$  through the end of  $s$ . We have filled in the first one for you.

KNOW P:	WANT R:	Additional info needed
(a) $x$ is the sum of $1..n$	$x$ is the sum of $1..100$	$n == 100$
(b) $x$ is the sum of $1..(n-1)$	$x$ is the sum of $1..100$	
(c) $x$ is the smallest char in $s[0..k-1]$	$x$ is the smallest char in $s[0..s.length()-1]$	
(d) $x$ is the no. of blanks in $s[0..k-1]$	$x$ is the no. of blanks in $s[0..]$	
(e) $x$ is the smallest char in $s[h..]$	$x$ is the smallest char in $s[0..]$	
(f) $x$ is the product of $k..n$	$x$ is the product of $1..n$	
(g) $b = \text{"no element of } h..k \text{ divides } x\text{"}$	$b = \text{"no element of } m..k \text{ divides } x\text{"}$	

4. Below is a precondition  $P$ , an assignment to a variable, and the same assertion  $P$  as a postcondition. At the place indicated, place a statement so that if  $P$  is true initially, it will be true afterward (as indicated). The statement can be in English, if you are not sure how to write it in Java, but make it a command to do something. In the exercises below,  $v$  is a Vector, and you can assume that its elements are of class Integer.

(a) <code>// {P: x is the sum of 1..n}</code>	(b) <code>// {P: x is the sum of h..100 }</code>
<code>// Put a statement here:</code>	<code>// Put a statement here:</code>
<code>n = n+1;</code>	<code>h = h - 1;</code>
<code>// { P: x is the sum of 1..n }</code>	<code>// { P: x is the sum of h..100 }</code>

## Lab 07. CS1110 Fall 2011. Practice with assertions and loops

```
(c) // { x is the minimum of v[0..k-1] }
    // Put a statement here

    k= k+1;
    // { x is the minimum of v[0..k-1] }
```

```
(d) // { x is the minimum of v[h..100] }
    // Put a statement here

    h= h - 1;
    // { x is the minimum of v[h..100] }
```

5. Start a new folder named lab07. Open the course web page in a browser, click on “Labs”, and download file Lab07.java into folder lab07. Open the file in DrJava. Class Lab07 contains two partially completed functions and two functions stubbed in just with return statements so that they compile.

Complete the functions in class Lab07. Each must contain a loop. Write one at a time, implementing the specification that we give you. Make sure each function is correct before proceeding to the next one. Do this by writing suitable calls in DrJava’s Interactions pane or making up a JUnit test class. Use enough different test cases so that you really are sure that the function is correct. If the function uses a String value, make sure that it works on an empty string (one whose length is 0). File Lab07.java contains additional comments.

You may not finish all four functions during the lab. Near the end of the lab, show an instructor or consultant what you have done. If necessary, complete the lab during the next week and show what you have done to the instructor or consultant then.

Below, we show you on paper the functions you will be writing.

### Function 1.

Complete the following function by filling in the underlined parts. Test your function to make sure it is right.

```
/** = “p is a prime.” i.e. p is at least 2 and is divisible by only 1 and itself */
public static boolean isPrime(int p) {
    if (p < 2)
        return _____;
    if (p == 2)
        return _____;

    // Return false if some integer in 2..p-1 divides p
    // inv: no integer in 2..k-1 divides p
    for (int k= ____; k != p; k= k+1) {
        if ( _____ )
            return false;
    }
    // {no integer in 2..p-1 divides p}
    return _____;
}
```

### Function 2.

Type **char** is a primitive type — see pp. 224–226 of the class text. Characters are represented by integers, and the cast `(int)c` yields the integer that represents `c`. So, some characters are represented by primes and others are not. Complete the function given on the next page.

Question: How can you check out the function? How do you know that it works? Explain your testing strategy in comments in your code.

## Lab 07. CS1110 Fall 2011. Practice with assertions and loops

```
/** = a String that contains each capital letter (in 'A'..'Z') whose representation is prime */
public static String primeChars() {
    String s= _____;
    // inv: s contains each capital in 'A'..'Z' whose representation is prime
    for (char c = _____; c <= _____; c= (char)(c+1)) {
        if ( _____ )
            s= s + _____;
    }
    // {s contains each capital in 'A'..'Z' whose representation is prime}
    return s;
}
```

### Function 3

Write a function with the following specification.

```
/** = number of times character c appears in String s */
public static int noOfTimes(char c, String s)
```

For example, since 'e' occurs twice in "where", the following statement stores 2 in x:

```
x= noOfTimes('e', "where");
```

Questions: what range of integers does the function process? What is the invariant for your loop? (The answers to both should be clear from your code, so your code should contain the invariant.)

### Function 4

Write a function with the following specification.

```
/** = number of times the characters in s1 appear in s2 */
public static int noOfTimes(String s1, String s2)
```

This function could be used as follows. The call

```
noOfTimes("aeiou", "Where is it?")
```

yields the number of times a vowel occurs in "Where is it?". Note that duplicates in s1 are to be counted as many times as they occur in s1. For example,

```
noOfTimes("aaab", "ac")
```

returns 3.

The body of this function should contain a for-loop whose repetend calls the previous function that you wrote (the other function noOfTimes) once for each character in s1.

Questions: what range of integers does the function process? What is the invariant for your loop? (The answers to both should be clear from your code, so your code should contain the invariant as a comment.)