Lab 07. CS100J Spring 08. Practice with assertions and loops

Goal of this lab

Practice with assertions and loops that process a range of integers. Part of this lab is 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 ≥ h–1. For example, 5..4 is OK but 5..3 is not allowed.

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

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

Range	Want	Assignment
(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 to that the range has 0 elements:	

3. Each line below contains an assertion P, which we know is 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. 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 smallest char in s[0..k–1]	x is smallest char in s[0..s.length()–1]	
(d) x is number of blanks in s[0..k–1]	x is number of blanks in s[0..]	
(e) x is smallest char in s[h..]	x is smallest char in s[0..]	
(f) x is product of k..n	x is product of 0..n	
(g) b = "no element of h..k divides x"	b = "no element of m..k divides x"	

4. Below is a precondition P, an assignment to a variable, and the same assertion P. 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) // {P: x is the sum of 1..n}

   // Put a statement here:

   n= n+1;
   // { P: x is the sum of 1..n }

(b) // { P: x is the sum of h..100 }

    // Put a statement here:

   h= h – 1;
   // { P: x is the sum of h..100 }

(c) // { x is the minimum of v[0..k–1] }

    // Put a statement here

(d) // { x is the minimum of v[h..100] }

    // Put a statement here

    k= k+1;

    // { x is the minimum of v[0..k–1] }

    h= h – 1;

    // { x is the minimum of v[h..100] }

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 with only return statements, so that they compile.

Complete the functions in class Lab07. Each will 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 make 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).

    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. Complete the lab during the next week and show what you have done to the instructor or consultant next week.

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

Function 1.
Complete the following function (filling in the underlined things). Test your function to make sure it is right.

```
/** = "p is a prime." i.e. is at least 2 and is divisible by only 1 and itself */
public static isPrime(int p) {
        if (p < 2)
                return _____;
        if (p == 2)
                return _____;
        // Return false if p is divisible by an integer in 2..k-1
        // inv: p is not divisible by integers in 2..k-1
        for (int k= ____; k != p; k= k+1) {
                if (_____ )
                        return false;
                k= k+1;
        }
        // {p is > 2 and not divisible by an integer in 2..p-1
        return _____;
}
```

Function 2.
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?

Lab 07. CS100J Spring 08. 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'..c-1 whose representation is prime
    for (char c = _____; c <= _____; c++) {
        if ( _____ )
                    s= s + _____;
    }
    // {s contains each capital in 'A'..'Z' whose rep is a 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= noTimes('e', "where");
```

Questions: what range of integers does the function process? What is the invariant for your loop?

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("aaa", "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?