COMS W1004              MW 4:10-5:25 PM

Columbia University                          Spring 2014

**Homework 4: Problem Set**

**March 31, 2014**

Your CUNIX ID: ami2119

Your Last Name: Iwamizu

Your First Name: Akiko

**Circle the range that includes your UNI:**

Group 1 (aa3473-am4051)  Group 8 (kea2134-lvt2107)

**Group 2 (ama2231-av2425)** Group 9 (lz2371-mry2109)

Group 3 (ay2289-cme2133) Group 10 (msv2121-pc2627)

Group 4 (cmh2194-emh2213) Group 11 (pfa2103-sc3719)

Group 5 (emm2224-hv2169) Group 12 (sch2148-tat2133)

Group 6 (hwk2106-jk3667) Group 13 (tb2498-zn2116)

Group 7 (jl4161-kdj2109)

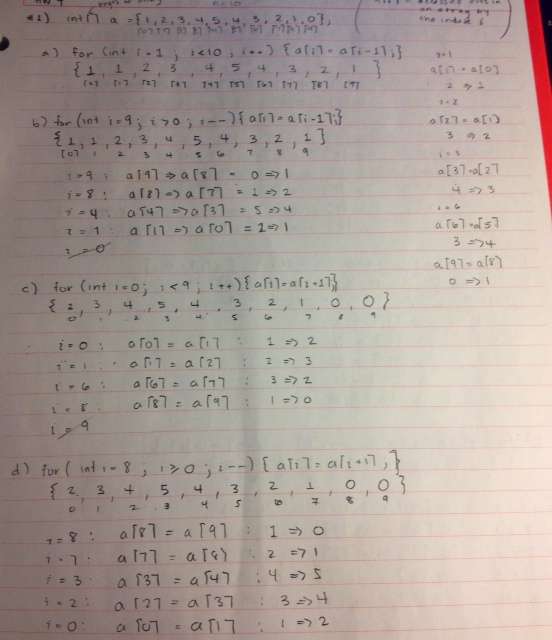
Do the following REVIEW EXERCISES (pages 362-366) in Chapter 7 of Big Java:

R7.3) Consider the following array: int[] a = {1,2,3,4,5,4,3,2,1,0};

What are the contents of the array a after the following loops complete?

1. for(int i=1; i<10; i++) {a[i] = a[i-1];}

(Note: I changed my answer for part (a) after revising my hw, so the work show in the picture isn’t acurate)

{1, 1, 1, 1, 1, 1, 1, 1, 1, 1}

1. for(int i=9; i>0; i--) {a[i] = a[i-1];}

{1, 1, 2, 3, 4, 5, 4, 3, 2, 1}

1. for(int i=0; i<9; i++) {a[i] = a[i+1];}

{2, 3, 4, 5, 4, 3, 2, 1, 0, 0}

1. for(int i=8; i>=0; i--) {a[i] = a[i+1];}

{2, 3, 4, 5, 4, 3, 2, 1, 0, 0}

1. for(int i=1; i<10; i++) {a[i] = a[i] + a[i-1];}

{1, 3, 6, 10, 15, 19, 21, 23, 24, 24}

1. for(int i=1; i<10; i = i+2) {a[i] = 0;}

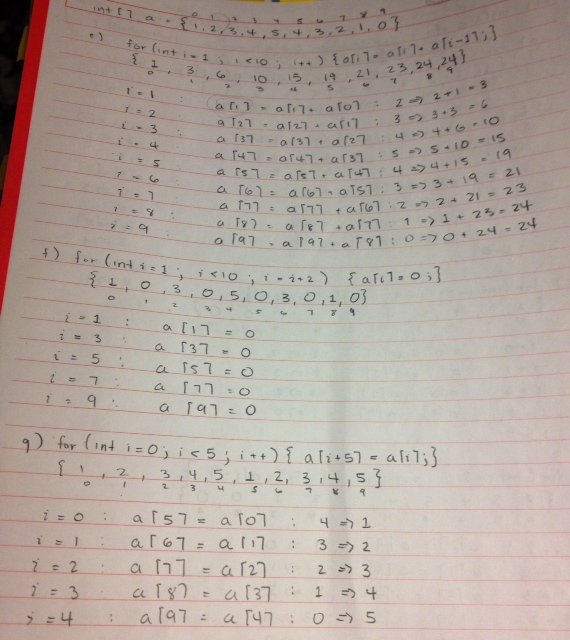
{1, 0, 3, 0, 5, 0, 3, 0, 1, 0}

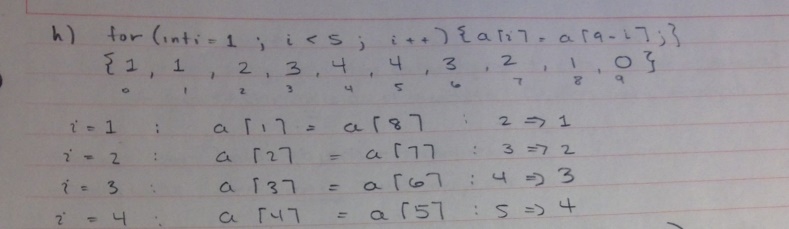
1. for(int i=0; i<5; i++) {a[i + 5] = a[i];}

{1, 2, 3, 4, 5, 1, 2, 3, 4, 5}

1. for(int i=1; i<5; i++) {a[i] = a[9 - i];}

{1, 1, 2, 3, 4, 4, 3, 2, 1, 0}





R7.5) Write Java code for a loop that simultaneously computes both the maximum and minimum of an array.

int largest = values[0];

int smallest = values[0];

for (int i = 1; i < values.length; i++)

{

if(values[i] > largest) largest = values[i];

if(values[i] < smallest) smallest = values[i];

}

Note: the loop starts at 1 because I initialized the *largest* and *smallest* with values[0].

R7.7) Write enhanced for loops for the following tasks:

1. Printing all elements of an array in *a single row* (Note: use print instead of println in this case), separated by spaces.

for(double elements : anArray)

{

System.out.print(elements + “ ”);

}

1. Computing the maximum of all elements in an array.

double largest = anArray[0];

for(double elements : anArray)

{

if(elements > largest) { largest = elements; }

}

1. Counting how many elements in an array are negative.

int counter = 0;

for(double elements : anArray)

{

if(elements < 0) { counter++; }

}

R7.9) Rewrite the following loops, using the enhanced for loop construct. Here, values is an array of floating-point numbers.

1. for(int i=0; i<values.length; i++) {total = total + values[i];}

for(float elements : values)

{

total = total + elements;

}

1. for(int i=1; i<values.length; i++) {total = total + values[i];}

for(float elements : values)

{

total = total + elements;

}

total = total – values[0];

1. for(int i=0; i<values.length; i++)

{

if(values[i] == target) {return i;}

}

int counter = 0;

for(float elements : values)

{

if(elements == target)

{

return counter;

}

counter++;

}

R7.22) A *run* is a sequence of adjacent repeated values. Give pseudocode for computing the length of the longest run in an array. For example, the longest run in the array with elements

1 2 5 5 3 1 2 4 3 2 2 2 2 3 6 5 5 6 3 1

has length 4.

int counter = 1;

int longestRun = 0;

for(int i = 0; i < values.length -1; i++)

{

while( values[i] == values[i+1] ) {

counter++;

}

if( counter > longestRun ){

longestRun = counter;

}

}

return longestRun;

R7.24) You are given two arrays denoting x- and y-coordinates of a set of points in the plane. For plotting the point set, we need to know the x- and y-coordinates of the smallest rectangle containing the points. How can you obtain these values from the fundamental algorithms in Section 7.3?

To begin, I would use the Maximum and Minimum algorithms to find the max and min values for both arrays. Once I find the max and min values of the x- and y-coordinates, I would print the combination of their coordinates to give the corners or “vertices” of the rectangle.

For example I would print the coordinates:

(max(x), max(y)) for the upper right hand vertex, (max(x), min(y)) for the lower right hand vertex, (min(x), max(y)) for the upper left hand vertex, and (min(x), min(y)) for the lower left hand vertex.

R7.31) True or false?

1. All elements of an array are of the same type.

**True**

1. Arrays cannot contain strings as elements.

**False**

1. Two-dimensional arrays always have the same number of rows and columns.

**False**

1. Elements of different columns in a two-dimensional array can have different types.

**False**

1. A method cannot return a two-dimensional array.

**False**

1. A method cannot change the length of an array argument.

**True**

1. A method cannot change the number of columns of an argument that is a two-dimensional array.

**True**

R7.32) How do you perform the following tasks with array lists in Java?

* 1. Test that two array lists contain the same elements in the same order.

To compare the two array lists I would use a for loop nested in another for loop to compare each value at each index. If the arrays contain or do not contain the same element in the same order, then I would use a Boolean to return the value true or false depending on the results.

* 1. Copy one array list to another.

To make a copy of an array list, I would construct the copy and pass the original list into the constructor.

Example:

ArrayList<String> newNames = new ArrayList<String>(names);

where “new ArrayList” constructs a new array list and passes the object reference “names” to newNames.

* 1. Fill an array list with zeroes, overwriting all elements in it.

Use a for loop to go through each index and set the value to 0 using something like

for(i=0; i < values.size(); i++){ values.set(i, 0); }

* 1. Remove all elements from an array list.

Use a for loop to go through each index and remove the values using something like

for(i=0; i < values.size(); i++){ values.remove(i); }

R7.33) True or false?

1. All elements of an array list are of the same type.

**True**

1. Array list index values must be integers.

**True**

1. Array lists cannot contain strings as elements.

**False**

1. Array lists can change their size, getting larger or smaller.

**True**

1. A method cannot return an array list.

**False**

1. A method cannot change the size of an array list argument.

**False**

R7.34) Define the terms regression testing and test suite.

Regression testing is when you keep old test cases and test every revision of a program against them. A test suite is a set of test cases for a program.