Homework 5

Due: April 15, 11:59 PM

Name: Daniel Carpenter

Student ID: 113009743

Submission Instructions: This Word document contains fillable fields where you can input your answers. (For example, you can input your name and ID in the fields above.) Please save the document regularly so your answers are not lost. After you complete the assignment, upload a copy to Gradescope in PDF format. A link to Gradescope can be found on the left side of the course Canvas page.

Important Note on Academic Integrity: This assignment should be completed individually. In recent semesters, improper collaboration on homework has led to multiple cases of plagiarism, where we receive identical or nearly identical submissions from two or more students. If you decide to discuss this assignment with other students before the deadline, make sure you first read the section of the syllabus on proper and improper collaboration. Additionally, you must include the names of these students below.

Names of Collaborators (if any):

Question 1 (10 points; 2 points per part): Write the signatures (only) for methods that perform the following tasks. Do not write the method body.

a) Create a perfect size array of integers of a given size that contains all of the values in a given perfect size array of integers that are between a given lower and upper bound (inclusive). Example: If the given array contained {1, 3, 5} and the lower bound was 2 and the upper bound was 6, the returned array would contain {3, 5}.

```
public static int[] boundedArray(int[] array, int lower, int upper)
```

b) Modify an oversize array of integers so that it contains only values within a given lower bound and a given upper bound (inclusive).

```
public static void adjBoundedArray(int lower, int upper)
```

c) Modify an oversize array of integers so that it contains only values from a second oversize array of integers that are within a given lower bound and a given upper bound (inclusive).

```
public static void takeArrayValues(int[] oversizeArray1, int[] oversizeArray2,  int lower, int upper)
```

d) Modify an oversize array of integers so that it contains only values from a second perfect size array of integers that are within a given lower bound and a given upper bound.

```
public static void takePerfectArrayValues(int[] oversizeArray, int[] perfectArray,  int lower, int upper)
```

e) Create a perfect size array of integers that contains only values from an oversize array of integers that are within a given lower bound and a given upper bound.

```
public static void takeArrayValues(int[] perfectArray1, int[] perfectArray2,  int lower, int upper)
```

Question 2 (10 points; 8 points for a), 2 points for b)): Write a method that takes two oversize arrays of String objects and creates an oversize array that contains exactly one copy of each of the objects that are in either of the two arrays.
For example: If one array contains {1, 4, 1, 6, 8} and is of size 5 and the other contains {2, 1, 4, 1, 6, 3} and is of size 2, If the resulting array had a capacity of ten and initially contained zeros, the resulting array should contain (1, 4, 6, 8, 2, 0, 0, 0, 0, 0} and be of size 5.
The method signature is below:
int unionWithoutDuplicates(int[] result, int[] first, int firstSize, int[] second, int secondSize)

```
Arrays.sort(first);
          Arrays.sort(second);

          for (int i = 0; i < second.length; ++i)
          {
               for (int j = 0; j < first.length; ++j)
               {
                    if (Arrays.binarySearch(first,second[j])
!= -1 && !(Arrays.binarySearch(result,second[j]) > -1))
                    {
                         result[j] = second[j];

                    }
                    else if
(Arrays.binarySearch(second,first[j]) != -1 &&
!(Arrays.binarySearch(result,first[j]) > -1))
                    {
                         result[j] = first[j];

                    }
               }
          }

System.out.println(Arrays.toString(result));
```

b) Write a code fragment that calls the method in a), as described in the example.

```
int[] result = new int[] {0,0,0,0,0,0,0,0,0,0};
int[] first = new int[] {1, 4, 1, 6, 8};
int firstSize = 5;
int[] second = {2, 1, 4, 1, 6, 3};
int secondSize = 2;

Int answer = unionWithoutDuplicates(result, first, firstSize, second, secondSize);
```

Question 3 (10 points; 5 per part): Trace the execution of the code fragments in the tables. Show all variable assignments, including the final values of loop variables, and use vertical space to indicate the simultaneous values of variables. (See the trace table guide for more information.) Note that the tables may contain more rows than you need.

a)

```
for (int n = 0; n < 5; ++n)
{
     double fraction = 1/2.0;
     int count = 0;
     for (double factor = fraction/2.0;
          count <= n && factor >= 0.05;
               factor= factor/2.0, ++count)
     {
          fraction = fraction + factor;
     }
     System.out.println(fraction);
}
```

n	count	fraction	factor
0	0	0.0	0.0
1	1	0.75	0.125
2	2	0.875	0.0625
3	3	0.9375	0.03125
4		0.9375	0.03125
5		0.9375	0.03125

```
b)
String word = new String("abcde");
String expanded = new String("");
char letter = word.charAt(0);
expanded = expanded + letter;
for (int index = 1; index < word.length(); ++index)
{
     int dup = index + 1;
     while (dup > 0)
     {
          expanded += "*";
          dup -= 1;
     }
     letter = word.charAt(index);
     expanded = expanded + letter;
}
```

word	index	dup	expanded	letter
abcde	1	2	“”	a
	2	1	a	b
	3	3	a*	c
	4	2	a**	d
	5	1	a**b*	e
		4	a**b**	
		3	a**b***	
		2	a**b***	
		1	a**b***c	
		5	a**b***c*	
		4	a**b***c**	
		3	a**b***c***	
		2	a**b***c****	
		1	a**b***c****d	
			a**b***c****d*	
			a**b***c****d**	
			a**b***c****d***	
			a**b***c****d****	
			a**b***c****d*****	
			a**b***c****d*****e	

Question 2 (5 points): An array of integers is constructed with the following line of code:

```
int[] sqr = {42, 12, 5, 9, 17, 23};
```

Suppose the array is sorted using **selection sort**. Show the steps of the algorithm in the trace table below.

Note: each row of the table should show the assignments made at the end of a single iteration of the outer loop. If the values of two elements are swapped, two values should appear in the row. If the value of an element is swapped with itself, only one value should appear. (See the sorting algorithm notes for examples.)

`sqr[0]`	`sqr[1]`	`sqr[2]`	`sqr[3]`	`sqr[4]`	`sqr[5]`
42	**12**	**5**	**9**	**17**	**23**
5		42			
	9		12		
		12	42		
			17	42	
				23	42

Question 3 (5 points): An array of integers is constructed with the following line of code:

```
int[] sqr = {42, 12, 5, 9, 17, 23};
```

Suppose the array is sorted using **insertion sort**. Show the steps of the algorithm in the trace table below. Each line should contain only one value that has been changed. Values that are not changed should not be shown on each line.

Note: use the version of the algorithm we discussed in class, not the version in your textbook. Show only a single assignment on each line. (See the sorting algorithm notes for examples.)

`sqr[0]`	`sqr[1]`	`sqr[2]`	`sqr[3]`	`sqr[4]`	`sqr[5]`	`temp`
42	**12**	**5**	**9**	**17**	**23**	
						12
	42					
12						
						5
		42				
	12					
5						
						9
			42			
		12				
	9					
						17
				42		
			17			
						23
					42	
				23		

Question 4 (5 points): Trace the execution of the code fragment in the table. Show empty Strings as “” (two double quotation marks).

```
StringBuilder source = new StringBuilder("rail");
StringBuilder result = new StringBuilder(" x ");

while (source.length() > 0)
{
     source.deleteCharAt(0);
     char c = 'z';
     if (source.length() > 0)
     {
          c = source.charAt(0);
     }

     result.insert(0, c);
     result.append(c);
}
```

source	Result	c
rail	x	z
ail	a x	a
il	a x a	z
l	ia x a	i
“”	ia x ai	z
	lia x ai	l
	lia x ail	z
	zlia x ail	
	zlia x ailz	

Question 5 (5 points): Show the ArrayList contents in the heap below. Indicate empty ArrayList elements by writing their indices in the identifier column and leaving their contents blank. (Don't forget to include and update the ArrayList size as well.) If the contents of a variable change, use a comma to separate the old and new values.

```
ArrayList<String> haiku = new ArrayList<String>(12);
haiku.add("Serious");
haiku.add("problem.");
haiku.add("All");
haiku.add(3, "shortcuts");
haiku.add(1, "error.");
haiku.add("have");
haiku.add("have");
haiku.add("disappeared.");
haiku.add("Screen.");
haiku.set(6, "computer");
haiku.add(haiku.get(0));
haiku.remove(2);
haiku.add("page");
haiku.remove("computer");
if (!haiku.contains("are")) {
     haiku.add("blank");
int ind = haiku.indexOf("page");
haiku.set(ind, "Mind.");
```

Heap

Identifier	Address	Contents
0	3000	Serious
1	3001	problem., error.
2	3002	All, problem., All
3	3003	Shortcuts, All, Shortcuts
4	3004	Shortcuts, have
5	3005	have, computer, disappeared
6	3006	have, computer, disappeared, Screen
7	3007	disappeared, Screen., Serious
8	3008	Screen., Serious
9	3009	Serious, , page, Mind.
10	3010	blank,
size	3011	0,1,2,3,4,5,6,7,8,9,9,10,9,10,10
	3012	

Question 6 (5 points; 1 per part for a), b), and d), 2 points for c)): A Bidi object is used to manage languages that are read both right to left and left to right (like Arabic and Hebrew).

a) Write a statement to import the Bidi class into a program.

```
java.text.Bidi
```

b) Write a statement that declares and constructs a Bidi object called letters that contains the String "abcde" and uses the flag that indicates that the text is read from left to right. This flag is a constant in the class.

```
int direction = Bidi.DIRECTION_LEFT_TO_RIGHT
Bidi letters = new Bidi("abcde", direction);
```

c) Write a code fragment that prints out "left to right" if the base direction for a Bidi object with reference text is left to right, "right to left" if the base direction is right to left, and "mixed" otherwise.

```
if (direction == Bidi.DIRECTION_LEFT_TO_RIGHT)
{
     System.out.println("left to right");
}
else if(direction == Bidi.DIRECTION_RIGHT_TO_LEFT)
{
     System.out.println("right to left");
}
System.out.println("mixed");
```

d) Print out "Requires Bidi" if the contents of a perfect sized character array with reference letters must use bidi analysis to be processed correctly.

```
if (Bidi.requiresBidi(letters, 0,letters.getLength()))
     {
          System.out.println("Requires Bidi");
     }
```

Question 7 (5 points): Write a method that performs insertion sort on an ArrayList of Integer objects. You may use only the size, get, add, and remove methods from the ArrayList class. The signature is below. You may use the code we developed in class as a starting point, if you wish to.

public static void sort(ArrayList<Integer> list)

```
list.add(120);
list.add(-1);
list.add(15);

// INSERTION SORT
for (int start = 1; start < list.size(); ++start)
{
          int temp = list.get(start);
          int index = start;

          while (index > 0 && temp < list.get(index - 1))
          {
               list.set(index, list.get(index - 1));
               --index;
          }
          list.set(index, temp);
}
```