Homework 5

Due: Nov. 14, 11:59 PM

Name:

Student ID:

Submission Instructions: This PDF 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. 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 (15 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)
{
    int factorial = 1;
    for (int factor = 1; factor <= n; ++factor)
    {
        factorial = factorial * factor;
    }
    System.out.println(n + "! = " + factorial);
}
```

n	factorial	factor

b)
```
for (int n = 0; n < 4; ++n)
{
    int tesseract = 1;
    for (int power = 0; power < 4; ++power)
    {
        tesseract *= n;
    }
    System.out.println(n + "^4 = " + tesseract);
}
```

n	tesseract	power

c)
```
String word = new String("wxyz");
String expanded = new String("");
for (int idx = 0; idx < word.length(); ++idx)
{
    int dup = idx + 1;
    while (dup > 0)
    {
        expanded += "-";
        dup -= 1;
    }
    char letter = word.charAt(idx);
    expanded = expanded.replace('-', letter);
}
System.out.println("Expanded word: " + expanded);
```

word	idx	dup	expanded	letter

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

```
int[] sqr = {25, 9, 4, 16, 36, 1};
```

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]
25	9	4	16	36	1

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

```
int[] sqr = {25, 9, 4, 16, 36, 1};
```

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

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
25	9	4	16	36	1	

Question 4 (5 points): Trace the execution of the code fragment in the table. Indicate the value of a StringBuilder object by writing the string that would be returned by calling toString. (Use double and single quotation marks to indicate strings and characters respectively.)

```
StringBuilder src = new StringBuilder("cat");
StringBuilder pal = new StringBuilder("o ");

while (src.length() > 0)
{
    char c = src.charAt(0);
    src.deleteCharAt(0);

    pal.append(c);
    pal.insert(0, c);
}
pal.reverse();
```

src	pal	c

Question 5 (10 points): Trace the execution of the program[1] in the memory diagram. Indicate empty ArrayList elements by writing their indices in the identifier column and leaving their contents blank. (Don't forget to include the ArrayList size as well.) If the contents of a variable change, use a comma, space, or some other delimiter to separate the old and new values.

```
public static void main(String[] args)
{
    ArrayList<String> haiku = new ArrayList<String>(12);
    haiku.add("a");
    haiku.add("reduces");
    haiku.add("your");
    haiku.add(3, "expensive");
    haiku.add(1, "crash");
    haiku.add("two");
    haiku.add("to");
    haiku.set(5, "computer");
    haiku.add(haiku.get(0));
    haiku.add("brick");
    haiku.add("simple");
    haiku.remove("two");
    haiku.remove(8);
    if (!haiku.contains("brick")) {
        haiku.add("stone");
    }
}
```

main Stack Frame

Identifier	Address	Contents
	700	
	701	
	702	

Heap

Identifier	Address	Contents
	3000	
	3001	
	3002	
	3003	
	3004	
	3005	
	3006	
	3007	
	3008	
	3009	
	3010	
	3011	
	3012	

[1] Haiku taken from the GNU Humor Collection: https://www.gnu.org/fun/jokes/error-haiku.en.html.

Question 6 (5 points; 1 per part): A linked list is a collection of data elements like an array.[2] The Java API includes the class LinkedList that implements this data structure. Use the LinkedList API documentation to answer the questions below. (Note: LinkedList is a generic class like ArrayList, so a type parameter must be given when constructing LinkedList objects or declaring LinkedList reference variables.)

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

b) Write a statement that constructs an empty LinkedList of Double object references. Assign the address of the list to a variable named "linkList." (Note: like an ArrayList, a LinkedList cannot store primitive data, only object references.)

c) Write a statement to add the number -273.15 to the list constructed in part (b). (Note: you can use autoboxing instead of explicitly constructing a Double object.)

d) Suppose some additional data is added to the list. Write a statement that prints the number of elements to the console.

e) Write a statement that removes the third element in the list and assigns it to a double variable named "third." (Note: you can use unboxing instead of calling the method doubleValue on the Double object.)

[2] Unlike an array, the elements may be stored in nonsequential memory locations. Each element is stored in a "node" that includes a reference to the next node in the list. The nodes can be located anywhere in memory, but a given node can only be accessed by iterating through the nodes that precede it in the list.

Question 7 (5 points): Write a method (signature and body) named "flipColumn" that takes a 2D boolean array and flips all the elements in a given column. For example, if the method is given the array

```
{{true, true, true}, {false, false, false}}
```

and the column index 1, it changes the array to

```
{{true, false, true}, {false, true, false}}.
```

Note that the method does not make a copy of the array.

You may assume that the given array is rectangular and not empty. You may also assume that the column index is greater than or equal to 0 and less than the number of columns.