CS1027A – Computer Science Fundamentals II

Fall 2021

Final Exam

Sunday, December 19, 2021

Multiple Choice		**45**
Longer Multiple Choice		**15**
Written Answers / Diagrams		**24**
Q51		7
Q52		7
Q53		10
Algorithm Design		**16**
Total		**100**

- Clearly write / bubble your name, student number, section number, and 3-digit exam code on the Scantron sheet and in the spaces below.
- Use an HB pencil on the Scantron sheet.
- Make sure phones and other electronic devices are turned off and put away.
- Do not flip this page until instructed to do so.

Student Name: ____

Student Number: ____

Section Number (001 or 002): ________________________________

Multiple Choice (45 questions x 1 mark each)

Select the **best option** for each question. The answers for this section **must** be filled in on your Scantron sheet. For questions that reference other classes (i.e. ArrayStack), please assume that the given code is in the same project as the referenced class(es) so that it will compile and run.

1. Indicate which of the following is an example of a non-linear collection.

    a. Stack
    b. Queue
    c. Tree
    d. Both a and b are true
    e. Both a and c are true

2. Local variables are stored in which part of memory?

    a. Static heap
    b. Dynamic heap
    c. Local drive
    d. Execution stack
    e. Both a and b are true

3. ~~The term "time complexity" is the measure of:~~

    a. ~~how many b[illegible] operatio[illegible]ed to be executed~~
    b. ~~the duration of t[illegible]m execution~~
    c. ~~the cloc[illegible]a of the comp[illegible]unning the program~~
    d. ~~how [illegible]ng it takes to program the al[illegible]rithm~~
    e. ~~None of the above~~

4. ~~Indicate which statement is true regarding the time complexity of the pop() method on a stack.~~

    a. ~~In Ar[illegible]k, pop() has a time [illegible]exity of O(1)~~
    b. ~~In ArrayStack, p[illegible] ha[illegible]e complexity of O(n)~~
    c. ~~In LinkedStac[illegible]() ha[illegible]ime complexity of O(1)~~
    d. ~~Both a[illegible] are true~~
    e. ~~Both b and c are true~~

5. How could you access the second element in a singly linked list, assuming front points to the first element of the linked list?

    a. front[1]
    b. front.getNext()
    c. front.getNode(1)
    d. front.getList[1]
    e. front.node[1]

6. Indicate what would be printed to the console if we called the following function with: System.out.println(run(15));

```
public static int run(int x) {
    if (x < 0) {
        return x;
    } else {
        return run(x-4);
    }
}
```

    a. -1
    b. 0
    c. 1
    d. 4
    e. 15

7. Indicate what would be printed to the console if we called the following function with: System.out.println(work(3));

```
public static int work(int x) {
    if (x < 7) {
        return 1 + work(x+1);
    } else {
        return 1;
    }
}
```

    a. 1
    b. 3
    c. 4
    d. 5
    e. 7

8. Which of the sorting algorithms perform a "divide and conquer" approach.

   a. Insertion sort
   b. Selection sort
   c. Quick sort
   d. Both a and b
   e. Both a and c

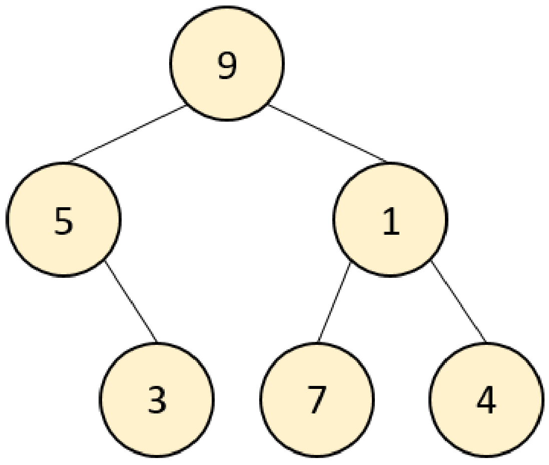
9. Indicate what is stored in the heap part of memory.

   a. Classes
   b. Interfaces
   c. Objects
   d. All of the above
   e. Both a and b

10. When performing a postorder traversal on a binary search tree, the last node visited is:

    a. The root node
    b. The leftmost node
    c. The rightmost node
    d. The node with the largest value
    e. It depends on the elements in the tree

Use the following diagram for Questions 11 to 15.



11. Determine the number of leaf nodes in the tree depicted in the diagram on page 4.

   a. 1
   b. 2
   c. 3
   d. 5
   e. 6

12. Determine the preorder traversal on the tree depicted in the diagram on page 4.

   a. 1 3 4 5 7 9
   b. 9 5 3 1 7 4
   c. 9 5 1 3 7 4
   d. 5 3 9 1 7 4
   e. 3 5 9 7 4 1

13. Determine the inorder traversal on the tree depicted in the diagram on page 4.

   a. 9 5 1 3 7 4
   b. 9 5 3 1 7 4
   c. 3 5 9 7 1 4
   d. 5 3 9 7 1 4
   e. 9 3 5 1 7 4

14. Determine the postorder traversal on the tree depicted in the diagram on page 4.

   a. 1 3 4 5 7 9
   b. 3 5 7 4 1 9
   c. 5 3 7 1 4 9
   d. 9 5 3 1 7 4
   e. 4 7 3 1 5 9

15. Determine the level order traversal on the tree depicted in the diagram on page 4.

   a. 1 3 4 5 7 9
   b. 9 5 3 1 7 4
   c. 5 3 7 1 4 9
   d. 3 7 4 5 1 9
   e. 9 5 1 3 7 4

16. Indicate the time complexity of the given method, doWork().

```
public static int doWork (int x) {
    if (x <= 1) {
        return 1;
    }
    return 1 + doWork(x / 2);
}
```

a. ~~O(1)~~
b. ~~O(log n)~~
c. ~~O(n)~~
d. ~~O(n * log n)~~
e. ~~O(n^2)~~

17. Based on the code provided in Question 16, what would be returned if we called doWork(32);

a. 1
b. 5
c. 6
d. 16
e. 32

18. What is the maximum number of activation records that would be simultaneously on the execution stack (the most at any given time) when calling doWork(8) (from Question 16) from the main method?

a. 2
b. 3
c. 4
d. 5
e. 6

19. What is the proper line for updating the rear in enqueue() in a Circular Array Queue?

a. rear = rear + 1;
b. rear = (rear + 1) % queue.length;
c. rear = (rear % queue.length) + 1;
d. rear = (rear - 1) % queue.length;
e. rear = (rear % queue.length) - 1;

20. Suppose we have the following array, arr, and we want to sort arr using an in-place **Insertion Sort**. Determine what would be in arr after 4 iterations of the sort.

4	9	7	1	5	2	3

a. [4, 9, 7, 1, 5, 2, 3]
b. [1, 2, 3, 4, 9, 7, 5]
c. [1, 4, 7, 9, 5, 2, 3]
d. [1, 2, 3, 4, 5, 7, 9]
e. [9, 7, 5, 4, 3, 2, 1]

21. Suppose we have the following array, arr, and we want to sort arr using an in-place **Selection Sort**. Determine what would be in arr after 4 iterations of the sort.

4	9	7	1	5	2	3

a. [4, 9, 7, 1, 5, 2, 3]
b. [1, 2, 3, 4, 9, 7, 5]
c. [1, 4, 7, 9, 5, 2, 3]
d. [1, 2, 3, 4, 5, 7, 9]
e. [1, 2, 3, 4, 5, 9, 7]

22. Select the line of code to properly fill in the blank to print all elements from the iterator.

```
public static void print (Iterator<String> iter) {
    ____________________
        System.out.println(iter.next());
    }
}
```

a. if (iter.next() != null) {
b. while (iter.next() != null) {
c. while (iter[i+1] != null) {
d. while (iter.hasNext()) {
e. for (int i = 0; i < iter.size(); i++) {

~~23. Indicate which of the following time complexities is the worst (i.e. requires the most operati~~ [illegible]

~~a. O(n)~~
~~b. O(~~ [illegible]
~~c.~~ [illegible] ~~(2^n)~~
~~d. O(n * log(n))~~
~~e. O(1)~~

~~24. The enqueue() method implemented with a singly linked list has a time complexity of _____ and the sam~~ [illegible] ~~od implemented w~~ [illegible] ~~oubly linked list has a time complexity of _____.~~

~~a. O(n), O(~~ [illegible]
~~b.~~ [illegible] ~~(n)~~
~~c.~~ [illegible] ~~(n), O(n)~~
~~d. O(1), O(log(n))~~
~~e. O(1), O(1)~~

25. Suppose we have class A with the method: public void sing(). We also have class B which inherits from A, and in class B there is the method: public void sing(String lyrics). These methods are:

a. overloaded
b. overridden
c. static
d. All of the above
e. Both a and b are true

26. Indicate what would happen if we tried to compile and run the following code fragment.

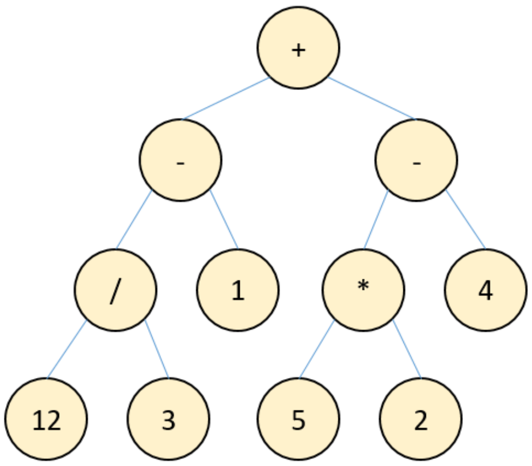
```
Object stack = new ArrayStack<String>();
stack.push("Happy Holidays");
```

a. Line 1 would cause a compilation error
b. Line 2 would cause a compilation error
c. Line 1 would cause a runtime exception
d. Line 2 would cause a runtime exception
e. The code would compile and run properly

27. Using the recursive Fibonacci method, rfib(), as shown in class, how many calls to rfib(2) would be made implicitly if we called rfib(5) to determine the 5th number in the Fibonacci sequence?

   a. 1
   b. 2
   c. 3
   d. 4
   e. 5

28. Evaluate the following expression tree.



   a. 3
   b. 7
   c. 9
   d. 12
   e. None of the above

29. Indicate what would happen if we tried to compile and run the following code fragment.

```
QueueADT<Integer> queue = new LinkedQueue<Integer>();
queue.enqueue(new Integer(25));
```

   a. Line 1 would cause a compilation error
   b. Line 2 would cause a compilation error
   c. Line 1 would cause a runtime exception
   d. Line 2 would cause a runtime exception
   e. The code would compile and run properly

30. A preorder traversal on a binary tree has a time complexity of:

    a. [illegible]
    b. ~~O(log n)~~
    c. ~~O(n)~~
    d. [illegible]
    e. ~~O(n^3)~~

31. Given the array, arr, depicted below, what would be returned from calling arr.length

T	P	W	A			

    a. 1
    b. 3
    c. 4
    d. 7
    e. It depends if this is a circular array or a standard array

32. Given the following Circular Array Queue, queue, and code fragment that references queue, indicate what would occur if the code was executed.

```
int v = front;
for (int t = 0; t < queue.length; t++) {
    System.out.print(queue[v]);
    v++;
}
```

S	F	B	N
	rear	front	

    a. It would print out: SFBN
    b. It would print out: BNSF
    c. It would print out: BN and then an exception would be thrown
    d. It would print out: BNSF and then an exception would be thrown
    e. An exception would be thrown immediately upon beginning execution

33. Indicate which collection(s) may have to be expanded to create additional space for new elements.

    a. Standard arrays
    b. Circular arrays
    c. Linked data structures
    d. All of the above
    e. Both a and b are true

34. How many push operations have to be performed during the stack-based Insertion Sort on an array of n elements which are already in ascending order?

    a. ~~1~~
    b. ~~log n~~
    c. ~~n~~
    d. ~~n log n~~
    e. ~~n^2~~

35. How many push operations have to be performed during the stack-based Insertion Sort on an array of n elements which are already in descending order?

    a. ~~1~~
    b. ~~log n~~
    c. ~~n~~
    d. ~~n log n~~
    e. ~~n^2~~

36. What would be printed out if the following code was executed (assume the toString() method in ArrayUnorderedList would print out each item from front to rear with spaces in-between)?

```
ArrayUnorderedList<Integer> list = new ArrayUnorderedList<Integer>();
for (int k = 12; k >= 5; k--) {
    list.addToFront(k);
}
System.out.println(list);
```

    a. 5 6 7 8 9 10 11 12
    b. 6 7 8 9 10 11 12
    c. 12 11 10 9 8 7 6 5
    d. 12 11 10 9 8 7 6
    e. The list would be empty

37. Consider the following variable declaration. What is the variable type of x.getElement()?

```
LinearNode<LinearNode<Integer>> x;
```

   a. Integer
   b. int
   c. LinearNode<Integer>
   d. LinearNode<LinearNode<Integer>>
   e. Both a and b are true

38. Indicate what would be printed out if the following code was executed.

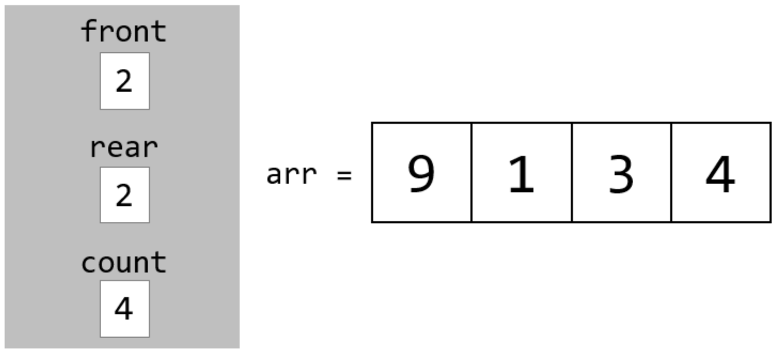
```
public class Reindeer {
    public static String name;
    public Reindeer (String n) {
        name = n;
    }
    public static void main(String[] args) {
        Reindeer[] deer = new Reindeer[] {
            new Reindeer("Comet"),
            new Reindeer("Vixen"),
            new Reindeer("Rudolph")
        };
        for (int r = 0; r < 2; r++) {
            System.out.print(deer[r].name + " ");
        }
    }
}
```

   a. Comet Vixen
   b. Comet Vixen Rudolph
   c. Vixen Rudolph
   d. Rudolph Rudolph
   e. Comet Comet

39. What is the maximum number of swaps required for the in-place Selection Sort on n elements?

   a. n-1 swaps
   b. n swaps
   c. n+1 swaps
   d. 2*n swaps
   e. n^2 swaps

40. Suppose we have the following Circular Array Queue, arr, and then we double the capacity to make room for more elements. How would the updated array be structured?



a. [3, 4, 9, 1, _, _, _, _]
b. [_, _, 3, 4, 9, 1, _, _]
c. [9, 1, 3, 4, _, _, _, _]
d. All of the above
e. Both a and b are true

41. Consider the following String array, arr, and code fragment that corresponds to the depicted array. What would be printed out from executing the following code?

D	E	C	E	M	B	E	R

```
int countE = 0;
int p = 0;

while (p < arr.length && arr[p].compareTo("E") != 0) {
    countE++;
    p++;
}

System.out.println("E count = " + countE);
```

a. E count = 0
b. E count = 1
c. E count = 3
d. E count = 7
e. E count = 8

42. Consider the following String array, arr, and code fragment that corresponds to the depicted array. What would be in list after executing the following code?

X	G	E	Q	B	H	K	A	T	W

```
ArrayOrderedList<String> list = new ArrayOrderedList<String>();
for (int p = 7; p > 3; p--) {
    list.add(arr[p]);
}
```

a. A B H K
b. A K H B
c. A K H B Q
d. B H K A
e. X G E Q

43. Consider the following line of code. Which of the options below is correct?

```
public class Foo {

    private LinearNode<String> b;

    public Foo (String x) {
        b = new LinearNode<String>(x);
    }

}
```

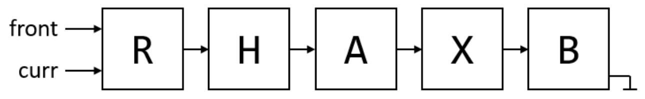
a. The variable x is stored in the heap, and the object referenced by b is temporarily on the execution stack
b. The variable x is temporarily on the execution stack, and the object referenced by b is stored in the heap
c. The variable x and the object referenced by b are stored in the heap
d. The variable x and the object referenced by b are temporarily on the execution stack
e. The class Foo is in the dynamic heap and the variable x is in the static heap

44. Determine the final value of q if the following code was executed.

```
ArrayStack<Object> stack = new ArrayStack<Object>();
stack.push("Merry Christmas");
stack.push(45);
stack.push(new Rectangle(7, 2));
stack.push("Happy Holidays");
stack.push(new Square(9));
stack.push("Happy New Year");
stack.push(21);
stack.push(new Rectangle(1, 5));

int q = 0;
while (!stack.isEmpty()) {
    Object x = stack.pop();
    if (x instanceof String) q++;
    else if (x instanceof Integer) q *= 2;
    else if (x instanceof Rectangle) q += 10;
    else if (x instanceof Square) q += 5;
}
```

a. 8
b. 48
c. 75
d. 85
e. 95

45. Examine the following linked list and code fragment that corresponds to the linked list. Determine what would be inside key after executing the following code.



```
ArrayOrderedList<String> key = new ArrayOrderedList<String>();
while (curr != null && curr.getElement().compareTo("T") < 0) {
    key.add(curr.getElement());
    curr = curr.getNext();
}
```

a. R, H, A
b. A, H, R
c. A, B, H
d. R, H, A, X, B
e. A, B, H, R, X

Multiple Choice (5 questions x 3 marks each)

This section is just like the above Multiple Choice section, but these questions are worth 3 marks each because they involve extra work to solve them.

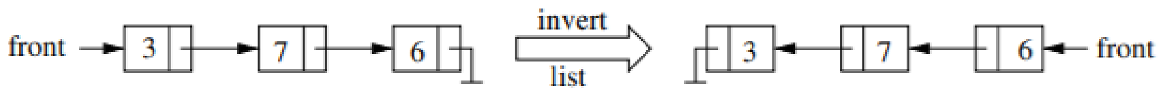
46. Encode the message below with a Caesar cipher using the repeating key: {4, 1, 3}. If a letter goes beyond Z, then wrap around to the start of the alphabet again (i.e. Z+1 = A)

    HAPPY HOLIDAYS

    a. IBQQZ IPMJEBZT
    b. LBSTZ KSMLHBBW
    c. LBSQZ IPMJEBZT
    d. LDTTC LSPMHECW
    e. GZOOX GNKHCZXR

47. Consider the following code fragment (with one line of code removed, shown with the blank line) and the diagram below that illustrates what the completed code fragment is supposed to accomplish: to invert a non-empty singly linked list. Select the line to fill in the blank in order to properly complete this code fragment.

```
LinearNode<Integer> curr = front;
LinearNode<Integer> next = curr.getNext();
LinearNode<Integer> prev = null;
while (curr != null) {
    ______________________________
}
front = prev;
```



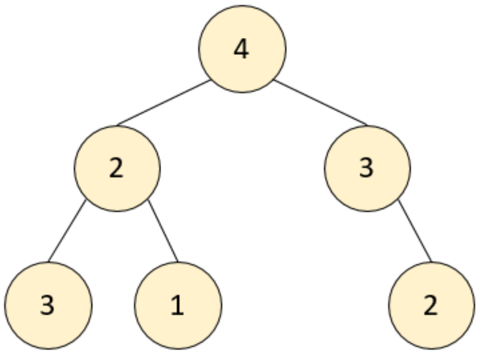
    a. next.setNext(prev); prev = curr; curr = next; if (next != null) next = next.getNext();
    b. front.setNext(prev); curr = next; next.setNext(prev); prev = curr;
    c. curr.setNext(prev); prev = curr; curr = next; if (next != null) next = next.getNext();
    d. next.setNext(curr); prev = curr; curr = next; if (next != null) next = next.getNext();
    e. prev = curr; curr = next; curr.setNext(prev); if (next != null) next = next.getNext();

48. Evaluate the following postfix expression:

7 2 10 2 / 1 + * 2 4 * - +

a. 3
b. 4
c. 10
d. 11
e. This is an invalid postfix expression

49. Consider the following binary tree and code fragment. Indicate the value returned from the method call: mystery(root); (assuming root is the root node of the tree depicted here)



```
public int mystery(BinaryTreeNode<Integer> n) {
    if (n == null) {
        return 0;
    } else {
        int v = mystery(n.getLeft()) + mystery(n.getRight());
        if (v > n.getElement()) return v;
        else return n.getElement();
    }
}
```

a. 4
b. 5
c. 6
d. 7
e. 15

50. Examine the following code and determine what would be printed to the console if the code was executed.

```
public static void main(String[] args) {

    int p = 5, f = 10;

    try {

        ArrayQueue<Integer> queue = new ArrayQueue<Integer>();

        for (int i = 1; i < 5; i++) {
            queue.enqueue(i*2);
        }

        for (int i = 1; i < 3; i++) {
            queue.enqueue(queue.dequeue());
        }

        while (p > 0) {
            f = queue.first() - p;
            for (int i = 0; i < f; i++) {
                queue.dequeue();
            }
            p--;
        }

        System.out.println("Done");

    } catch (NullPointerException e) {
        System.out.println("Found a bug");
    } catch (ArrayIndexOutOfBoundsException e) {
        System.out.println("f = " + f);
    } catch (EmptyCollectionException e) {
        System.out.println("p = " + p);
    }

}
```

a. f = -1
b. p = -1
c. p = 4
d. Found a bug
e. Done

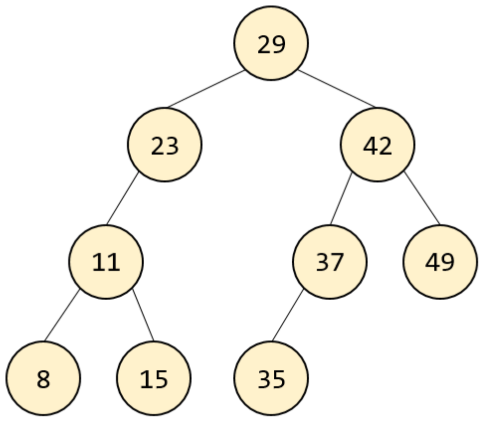
Written Answer / Diagrams

51. [7 marks] Consider the array arr = [11, 5, 17, 9, 13, 20, 14]. Trace through the Quick Sort algorithm on this array. Use the first element as the pivot at each level. Show all your work including the current array being handled, the pivot, and each of the sub-arrays for smaller, equal, and larger. Do not sort or re-order anything without showing your work for that step! Draw arrows or use numbers to help identify the order of the steps.

52. [7 marks] Examine the following code fragment and hand-trace its execution. As each method is called or is finished executing, show what the execution stack would look using the stack diagrams below. Include all the relevant information required for each activation record within the execution stack. Note that System.out.print() calls don't have to be shown in the diagrams.

```
public class Snowman {

    public Snowman (int baseSize) {
        boolean status = addBall(baseSize);
        createFace("coal", "button", "corn cob");
    }

    public boolean addBall (int size) {
        if (size <= 4) return true;
        else {
            System.out.print("Adding snowball of size " + size + "\n");
            return addBall(size-2);
        }
    }

    public void createFace (String eyes, String nose, String pipe) {
        System.out.print("Drawing snowman's face with: " + eyes);
        System.out.print(", " + nose + ", and " + pipe);
    }

    public static void main (String[] args) {
        Snowman frosty = new Snowman(10);
    }
}
```

53. Examine the following binary search tree. For each of the parts below, explain the steps that would need to be taken to try to find the nodes containing the target elements. If a node does not exist, explain where it would need to be added, after showing the steps to get to that spot.



a) [5 marks] find(35);

b) [5 marks] find(18);

Algorithm Design

54. [16 marks] Write two methods, as described below, in pure Java or Java-like pseudocode.

    The first method, isPalindrome(), must take in a CircularArrayQueue of Strings and check if the elements in that queue form a palindrome, i.e. if we looked at each of the elements from front to rear, they would also be the same elements from rear to front. In other words, element 1 = element n, element 2 = element n-1, and so on. For example, a queue with the elements {"R", "A", "D", "A", "R"} is a palindrome so the method should return true. You can use recursion or iteration to accomplish this. You **cannot** use a Stack, List, Array, or any other kind of data structure or collection to help with this. You also cannot create additional queues. Large penalties will be applied if you use a Stack or any other collection. Note that the original queues do not have to be preserved after calling this method.

    The second method is queueLog() and it must take in an Iterator of such queues and you must loop through them to count how many of the queues are palindromes out of the total number of queues. The method must print out a message of the form "2 of 3 queues are palindromes".

    The two method signatures must be exactly as shown here:

    ```
    public static boolean isPalindrome (CircularArrayQueue<String> queue)
    public static void queueLog (Iterator<CircularArrayQueue<String>> iter)
    ```