

15-121
Fall 2018 Midterm Exam
October 16, 2018

Name:

Andrew ID:

- You may not use any books, notes, or electronic devices during this exam.
- Show your work on the exam to receive credit.
- You may complete the problems in any order you'd like; you may wish to start with the free response problems, which are worth most of the credit.
- All code samples run without crashing unless we state otherwise.
- Assume any imports are already included as required.

Don't write anything in the table below.

Question	Points	Score
1	10	
2	5	
3	15	
4	15	
5	10	
6	15	
7	30	
8	0	
Total:	100	

2. (5 points) **Code Tracing**

Indicate what the code will print. Place your answer (and nothing else) in the box below the code.

```
public class CT1 {  
  
    public static int mystery(int n, int m) {  
        while (n >= m) {  
            n = n - m;  
        }  
        return n;  
    }  
  
    public static void main(String[] args) {  
        int num = mystery(24, 5);  
        System.out.println(num);  
    }  
}
```

3. (15 points) **Big-Oh**

Determine the big-oh runtime of each of the following, in terms of N , the length of the array. Write your answer, and nothing else, in the box next to each function.

```
public static int func1(int[] a, int b) {  
    for (int i = 0; i < a.length; i++) {  
        if (a[i] == b) {  
            return i;  
        }  
    }  
    return -1;  
}
```

```
public static void func2(int[] a, int b, int c) {  
    if (b < a.length) {  
        a[b] = c;  
    }  
}
```

```
public static void func3(int[] a, int b) {  
    int t = func1(a, b);  
    while (t != -1) {  
        func2(a, t, 0);  
        t = func1(a, b);  
    }  
}
```

4. Linked List Memory Diagram

Consider the following program that creates a linked list. You may assume that the `ListNode` class exists and was defined as in class. (If you have forgotten it, you can find a copy in the handout.)

```

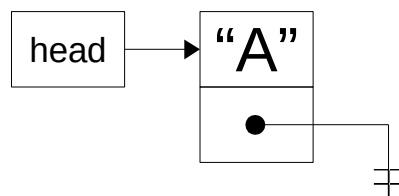
1  public class LinkedListDiagram {
2
3      public static void main(String[] args) {
4          ListNode head = null;
5          ListNode a = null;
6          ListNode b = null;
7          ListNode t = null;
8
9          a = new ListNode("A");
10         head = a;
11
12         b = new ListNode("B");
13         b.next = a;
14         head = b;
15         head.next.next = new ListNode("C");
16
17         t = head;
18         while (t.next != null) {
19             t = t.next;
20         }
21         t.next = new ListNode("D");
22
23         t = a;
24         a = a.next;
25         t.next = new ListNode("E");
26         t.next.next = a;
27     }
28 }

```

Starting from the head, draw the state of the linked list after the execution each specified line of code. The first one is done for you.

Note: There is an extra copy of this code in the Reference Handout provided to you with this exam. (So that you don't need to keep flipping pages...)

(a) After Line 10



(b) (5 points) After Line 15

(c) (5 points) After Line 21

(d) (5 points) After Line 26

5. (10 points) **Free Response: Word Length**

Write the public method `wordsCount` which, given an `ArrayList` of strings and an integer `len`, returns the number of words in the `ArrayList` that have length `len`.

For example, assuming an `ArrayList` named `arr` contains the strings `"a"`, `"bb"`, `"c"`, `"ddd"` then...

`wordsCount(arr, 1)` returns 2

`wordsCount(arr, 3)` returns 1

`wordsCount(arr, 4)` returns 0

6. (15 points) **Free Response: Inheritance**

Consider the following class:

```
public class Pet {  
    private String name;  
    private String owner;  
  
    public Pet(String name, String owner) {  
        this.name = name;  
        this.owner = owner;  
    }  
  
    public String toString() {  
        return this.name + " is owned by " + this.owner;  
    }  
}
```

Write a new class, `Cat`, that is-a `Pet`. Your `Cat` class should function properly when called as follows:

```
Pet myPet = new Cat("Fluffy", "Ahmed", true);  
System.out.println(myPet);  
// The previous line prints: Fluffy is owned by Ahmed and likes fish  
  
Pet otherPet = new Cat("Mittens", "Shaikha", false);  
System.out.println(otherPet);  
// The previous line prints: Mittens is owned by Shaikha and hates fish
```

Note: Your solution *must* properly apply the principals of inheritance. Writing a `Cat` class that does not properly inherit from `Pet` and make use of `Pet`'s methods and data will not receive points.

Write your answer on the next page. (Do not write your answer on this page.)

Space for answer to Question 6.

7. Free Response: Number Verification

We consider a sequence of numbers to be *jurbish*¹ if every number in the sequence is the sum of the previous three numbers. (Except the first three numbers, of course.)

Consider the following sequence:

1 3 4 8 15 27 50

This sequence is jurbish because $8 = 1 + 3 + 4$, $15 = 3 + 4 + 8$, $27 = 4 + 8 + 15$, and $50 = 8 + 15 + 27$.

- (a) (15 points) Write the function `isJurbish` which, given an array of integers as an argument, returns `true` if the sequence is jurbish and `false` otherwise.

For example:

```
isJurbish([1, 3, 4, 8]) returns True
isJurbish([1, 3, 4, 8, 15, 27, 50]) returns true
isJurbish([1, 3]) returns false
isJurbish([1, 3, 4]) returns false
isJurbish([1, 3, 4, 7]) returns false
isJurbish([1, 3, 4, 7, 14, 25, 46]) returns false
```

¹This is a fake term designed for this problem.

- (b) (15 points) Write the function `isFileJurbish` which, given a filename, returns `true` if every line in the file contains a jurbish sequence of integers and `false` otherwise.

For example, if `input.txt` contains:

```
1 3 4 8
1 3 4 8 15 27 50
```

Then `isFileJurbish("input.txt")` returns `true`

However, if `input.txt` contains:

```
1 3 4 8
1 3 4 8 15 27 49
```

Then `isFileJurbish("input.txt")` returns `false`

Hints:

- You should call `isJurbish`. You can assume it works even if yours does not.
- You can convert a string to an integer using `Integer.parseInt(String s)`. For example, calling `Integer.parseInt("15")` returns `15`.
- You may assume you can use the `getFileScanner(String filename)` method given in both the notes and the homework.

Additional Space for Answer to Question 7

8. (3 points (bonus)) **Extra Credit: Reasoning over Code**

Determine an appropriate argument to pass to `ROC1` that causes it to return `true`. Write your answer (the value for `s`), and nothing else, in the box below the code.

```
public static boolean ROC1(String s) {  
    if (s.length() % 2 == 1) {  
        return false;  
    }  
  
    String a = "";  
    String b = "";  
    for (int i = 0; i < s.length(); i += 2) {  
        a += s.charAt(i);  
        b += s.charAt(i + 1);  
    }  
  
    int aI = Integer.parseInt(a);  
    int bI = Integer.parseInt(b);  
  
    if (aI < 100 || bI < 100) {  
        return false;  
    }  
  
    return (aI + bI) == 567;  
}
```