This is a closed book exam. You have 75 minutes to complete the exam. There are 5 questions on this examination. The point values for the questions are shown in the table below. Your answers should fit in the space provided in the exam booklet. Paper for scrap work will be made available during the examination.

NAME: __________________________________________

LEcTure sECTION:      __ 09.00     __ 10.00

<table>
<thead>
<tr>
<th>Question</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>25</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

I have neither given nor received aid on this examination.

Signature: ________________________________
Question 1

Write Java code to implement the two methods described below. Your solutions should not use any built-in String methods that do substring replacement (i.e., replace and replaceAll). You may use other String methods including indexOf, substring, etc.

a) Define a method removeLetter that takes two parameters, word and letter, and returns the String obtained by removing one copy of letter from word. You may assume that letter is a String of length one. For example, removeLetter("mississippi", "p") would return "mississipi".

private String removeLetter( String word, String letter ) {
  int pos = word.indexOf( letter );
  if ( pos == -1 ) {
    return word;
  } else {
    return word.substring( 0, pos ) + word.substring( pos+1 );
  }
}
b) Define a method `removeAllLetter` that takes two parameters, `word` and `letter`, and returns a `String` obtained by removing all copies of `letter` from `word`. You may assume that `letter` is a `String` of length 1. For example, `removeAllLetter("mississippi","s")` would return "miiippi". You may use the `removeLetter` method from part (a) in your definition of `removeAllLetter` even if you are not confident your `removeLetter` implementation is correct.

```java
private String removeAllLetter( String word, String letter ) {
    while ( word.contains( letter ) ) {
        word = this.removeLetter( word, letter );
    }
    return word;
}
```
Question 2

Consider the code shown below. Much of it should look quite a bit like code you wrote as part of this week’s lab.

```java
public class StringList {
    private boolean empty = true;
    private String first;
    private StringList others;

    public StringList() {
    }

    public StringList( String text, StringList existing ) {
        empty = false;
        first = text;
        others = existing;
    }

    public String toString() {
        if ( empty ) {
            return "";
        } else {
            return others.toString() + first + " ";
        }
    }

    public int R() {
        if ( empty ) {
            return 0;
        } else {
            return first.length() + others.R();
        }
    }

    public StringList T( String target ) {
        if ( empty ) {
            return new StringList();
        } else if ( first.startsWith( target ) ) {
            return others;
        } else {
            return new StringList( first, others.T( target ) );
        }
    }

    public StringList V() {
        if ( empty ) {
            return new StringList();
        } else {
            return new StringList( first, others );
        }
    }
}
```
Suppose that we have executed the assignment:

```java
StringList words = new StringList( "greater",  
    new StringList( "than" ,  
        new StringList( "all" ,  
            new StringList( "the" ,  
                new StringList( "parts",  
                    new StringList()  
                )  
            )  
        )  
    );  
```

Indicate the String or int value that would be produced by each of the following expressions. If possible, also provide a brief description of what the first method invoked in each expression does in general to a StringList.

a) `words.toString()`

```
parts the all than greater
```

*The method converts the list into a String containing all of the Strings in the list in reverse order followed by spaces.*

b) `words.R()`

```
22
```

*The method determines the sum of the lengths of all the strings in a StringList.*

c) `words.V().toString()`

```
part the all than greater
```

*The method makes a copy of the first object in the StringList so that the resulting StringList looks just like the original.*

d) `words.T( "all" ).toString()`

```
parts the than greater
```

*The T method returns a StringList forming a copy of the original StringList with all Strings that contain the specified String removed.*
**Question 3**

Imagine that you would like to efficiently transmit the titles of songs and albums by the Irish folk rock band Clannad, such as 

*Anam*  
*An Gleann*  
*Macalla*  

(meaning *Soul, The Valley, and Echo*, respectively).

The table below shows the counts of the letters A, C, G, L, M, and N in the small sample of Clannad’s songs listed above.

<table>
<thead>
<tr>
<th></th>
<th>C</th>
<th>G</th>
<th>M</th>
<th>L</th>
<th>N</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

One possible Huffman tree that can be derived from the table above is:

![Huffman tree](image)

a. Show the binary sequence that would be used to encode the title “Macalla” using the tree above. (i.e. In the table below, fill in the correct binary code for each letter in the word.)

<table>
<thead>
<tr>
<th>M</th>
<th>A</th>
<th>C</th>
<th>A</th>
<th>L</th>
<th>L</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>1</td>
<td>0000</td>
<td>1</td>
<td>010</td>
<td>010</td>
<td>1</td>
</tr>
</tbody>
</table>

b. How many bits are needed to encode “Macalla”?  

16
c. Note that the length of the longest codeword given by the tree on the previous page is 4. Derive another Huffman tree from the table of letter occurrences above, where the length of the longest codeword represented by the tree is 5.

![Huffman Tree Diagram]

d. Since there are 6 symbols in the alphabet we’re considering (A, C, G, L, M, and N), a fixed-length code would require that each symbol be represented by three bits. Is there a Huffman tree for the letters and occurrence counts above for which the longest codeword would be 3? If so, show the tree. If not, briefly explain why not.

**NO.** The alphabet involved is small enough that you can simply argue that following the Huffman algorithm, no matter which choice you make at each point that a choice is possible because there is more than one symbol/subtree of minimal weight, you get a tree of height at least 4. Alternately, one can argue that a tree of height 3 with 6 leaves must have 4 paths of length 3 and two of length 2. Assigning the more common letters (N and A) to the shorter codes results in a code that requires 43 bits to encode all of the song names. Any of the Huffman trees considered only require 42 bits. Thus, a tree of height 3 cannot be optimal so it cannot be a Huffman tree.
Question 4

Suppose that computers A, B, C, and D are all connected to an Ethernet as shown below so that the length of the cables between any pair of computers is equal to \( L/2 \), where \( L \) is the maximum allowed length between any pair of computers on an Ethernet. Note that with this configuration, the distance between any pair of computers in the group A, B, and C is \( L \).

![Diagram of computer connections](image)

a) Suppose that A becomes ready to send a packet while D is already transmitting. Approximately how much time will elapse between the point when C hears the last bit of D’s packet and the first bit of A’s?

It takes time \( L/2c \) for D’s last bit to reach both A and C. Because Ethernet is 1-persistent, A will start sending soon after it senses this last bit because that is when it will detect that the network is idle. Once A transmits its first bit, it takes \( L/2c \) for it to reach D and another \( L/2c \) for it to continue on to B and C. So, C will detect the signal \( L/c + i \) time units later where \( i \) is the amount of time it takes to detect that the network is idle. (Although we have not discussed it in class, the Ethernet protocol places a minimum on \( i \). A computer must wait 96 bit transmission times before concluding the network is idle.)

b) Suppose that D becomes ready to send a packet while A is already transmitting. Approximately how much time will elapse between the point when C hears the last bit of A’s packet and the first bit of D’s?

close to 0. It takes time \( L/2c \) for A’s last bit to reach D and \( L/c \) time for A’s last bit to reach C. D will start sending soon after it senses the last bit from A because that is when it will detect that the network is idle. Once D transmits its first bit, it takes \( L/2c \) for it to reach C. So, C will detect the signal from D just a little more than \( L/c \) time units after A sent its last bit, which is essentially the same time it will receive D’s first bit.

c) Suppose that A, B and C all try to send packets containing 640 bytes (i.e., 5120 bits or 10 times the minimum packet size) at exactly the same time. All three computers will detect the collision simultaneously and then choose random delays according to the binary exponential backoff algorithm. Assume that after it delays, A is the first computer to successfully transmit and that no additional collisions have occurred. What random delay must each of the three computers involved have chosen given that A transmits successfully? Briefly explain your answer.

After the collision, all three machines will have to choose a delay value of 0 or 1. If A is going to succeed, its choice must be different from the choice made by B and C. Since there are only two choices, this means B and C must choose the same delay. Since this would lead to a collision if B and C choose the shorter delay, A must choose the delay of 0 and B and C must delay for one slot time. By the time B and C finish delaying, A will be successfully sending so B and C will follow the rules of carrier sense and wait for A’s transmission to complete.
d) Now, assume that A actually had two packets to send, so that A, along with B and C, will try to transmit again as soon as A's first transmission completes. This will lead to a second collision which all three computers will detect simultaneously. Again, all three computers will choose random delays. From what range of backoff values will each of the three computers choose? What is the probability that B is the first to transmit successfully during this second round of competition without further collisions? Explain briefly.

Since A just complete a successful transmission, it will choose to delay 0 or 1 slots after the collision. For B and C, on the other hand, this will be the second collision experienced so they will choose between delays of 0, 1, 2, and 3. To succeed, B must send before A, so B must choose a delay of 0 and A must choose a delay of 1. C must delay more than B, so it may choose 1, 2, or 3. The probability for each machine is determined by the number of slots it gets to choose from. B will choose 0 with probability 1/4, A will choose 1 with probability 1/2 and C will choose a slot other than 0 with probability 3/4 = (1-1/4). The probability all these events will occur is just the product of these probabilities: 1/4x1/2x3/4 = 3/32.

e) Suppose that A, B, and C each have many 640 byte packets to send so that the collision resolution process repeats many, many times. Assume that the network transmission rate is 10 megabits per second and that network monitoring shows that the network is operating at 90% efficiency and that the three computers are each able to send roughly equal numbers of packets. How many packets per second is A able to send? Don’t bother getting out your calculator for this question. A clear formula will be a better answer than an actual value.

\[
(0.9 \times 10^7 \text{ bits/second}) / (3 \text{ computers} \times 640 \text{ bytes/packet} \times 8 \text{ bits/byte}) =
\]

\[
(0.9 \times 10^7 \text{ bits/second}) / (3 \text{ computers} \times 5120 \text{ bits/packet}) =
\]

\[
0.9 \times 10^7 / 15360 \text{ packets per computer per second}
\]

f) Consider a network identical to that considered in (d) except that A, B, and C each have many 64 byte (i.e. 512 bit) packets to send. What value would you predict for the efficiency of this network based on the measured efficiency of the network described in part (e) and the formula from the paper by Metcalfe and Boggs shown below?

\[
\text{efficiency} = \frac{P/R}{WT + P/R} = \frac{1}{WT/(P/R) + 1}
\]

As usual, justify your answer.

The key to this question is to realize that the size of the packets being transmitted on this network will not change the amount of time wasted because of collisions and idle slots between successful transmission. As a result, of the four variable P, R, T and W used to predict the efficiency of a network, only P changes between this question and the preceding question. Plugging the measured value of the efficiency from part (e) into the second form of the efficiency equation gives where P is 5120.

\[
0.9 = \frac{1}{WT/(P/R) + 1} \Rightarrow WT/(P/R) = 1/9
\]

Now, if P' is 512, we know that P/10 = P' and therefore that WT/(P'/R) = WT/(P/10R) = 10/9. Plugging this into the efficiency formula gives

\[
\frac{1}{WT/(P'/R) + 1} = \frac{1}{19/9} = 0.47
\]
Question 5

The program on the next page draws five buttons in a window. These buttons -- button1, button2, button3, button4, and button5 -- are labeled “CIRCUMFERENCE”, “EQUALS”, “PI”, “TIMES”, “DIAMETER”, respectively, as shown below. Pressing the buttons changes the labels of various buttons.

Suppose that the buttons named button1, button2, button3, button4, and button5 are clicked in sequence. Indicate the labels on each of the buttons after the buttonClicked method is invoked in response to each of the five clicks in this sequence.

i) After clicking button1

```
button1  button2  button3  button4  button5
CIRCUMFERENCE  EATS  CAKE  PLUS  DIAMETER
```

ii) After clicking button2

```
button1  button2  button3  button4  button5
VOLUME  EATS  CAKE  PLUS  RADIUS^2
```

iii) After clicking button3

```
button1  button2  button3  button4  button5
AREA  EATS  CAKE  PLUS  RADIUS^2
```

iv) After clicking button4

```
button1  button2  button3  button4  button5
AREA  EQUALS  CAKE  TIMES  RADIUS^2
```

v) After clicking button5

```
button1  button2  button3  button4  button5
AREA  EQUALS  PI  TIMES  RADIUS^2
```
public class ExamButtons extends GUIManager {

    private JButton button1, button2, button3, button4, button5;

    public ExamButtons() {
        this.createWindow(600, 150);
        button1 = new JButton("CIRCUMFERENCE");
        button2 = new JButton("EQUALS");
        button3 = new JButton("PI");
        button4 = new JButton("TIMES");
        button5 = new JButton("DIAMETER");
        contentPane.add(button1);
        contentPane.add(button2);
        contentPane.add(button3);
        contentPane.add(button4);
        contentPane.add(button5);
    }

    public void buttonClicked( JButton which ) {
        button4.setText("PLUS");
        if ( which == button1 ) {
            button3.setText("CAKE");
            button2.setText("EATS");
        }
        if ( which == button2 ) {
            button1.setText("VOLUME");
        }
        if ( which == button3 ) {
            button1.setText("AREA");
            if (button3.getText().equals("PI")) {
                button3.setText("ICE CREAM");
            }
        }
        if ( which == button4 ) {
            if (button3.getText().equals("ICE CREAM")) {
                button2.setText("SLURPS");
            } else {
                button2.setText("EQUALS");
            }
        }
        if ( which == button5 ) {
            button4.setText("TIMES");
            if (button3.getText().equals("CAKE")) {
                button3.setText("PI");
            } else if ( which == button2 ) {
                button5.setText("RADIUS^2");
            }
        }
    }
}