Question 1.

Consider the mystery code below:

```java
int N = 6;
int result = 1;
while (N > 1) {
    result = result * N;
    N = N - 1;
}
textbox.append("" + result + "\n");
```

(a) What is appended to the textbox?
(b) How many times is the \(N > 1\) condition evaluated?
(c) If the first line was \(\text{int N = 0}\), then what would be written to the text box?
(d) If the first line was \(\text{int N = 0}\), then how many times would the \((N > 1)\) condition be evaluated?
(e) What is the mathematical name for the function of \(N\) that is computed?

Question 2.

A strand of bacterial messenger RNA can be encoded as a long string of letters representing the four possible bases: adenine (A), guanine (G), cytosine (C), and uracil (U). For example, one short RNA strand is:

```
UUACAUGUUAACGGACUAGGACGGAUGACACCAUAGAA
```

Only certain parts of an RNA strand are responsible for protein synthesis — the rest of the strand is unused while building proteins. The subsequences that are used for protein synthesis can be identified by the fact that they always begin with the triple of bases AUG, known as the start codon and end with the triple UAG, known as the stop codon. (Other stop codons exist, but we will ignore them.) The following is the example from above, rewritten to delineate the important subsequences from the junk:

```
UUAC
AUG
UUACGGAC
UAG
GACGG
AUG
ACACCA
AUG
AA
```

Real strands of RNA contain many thousands of bases, and the sections encoding proteins are much longer. Suppose we wish to write a segment of Java code that extracts the sequences of bases between adjacent start and stop codons in a form that would make it easy to display the results. We will assume that before our code is executed, other parts of the program will associate the variable `rna` with a `String` composed only of the letters A, G, C, and U, and at the end our code should associate the variable `result` with a `String` composed of just those subsequences of the original `String` that were preceded by a start codon and followed by a stop codon. Each sequence extracted should be separated from the next sequence by a newline character. For example, if the variable `rna` is initialized as

```
String rna = "UUACAUGUUAACGGACUAGGACGGAUGACACCAUAGAA";
```

then the desired code should associate the `String`

```
"UUACGGAC\nACACCA"
```

with the variable `result`. For simplicity, we will assume that a sequence never ends in the middle of a coding region, so every AUG will be matched by a UAG.

Here is one, correct version of the desired code:

```java
int start = rna.indexOf("AUG");
String result = "";
while (start >= 0) {
    int end = rna.indexOf("UAG", start);
    result = result + rna.substring(start + 3, end) + "\n";
    start = rna.indexOf("AUG", end + 3);
}
```
Below, you will find four code fragments that are similar but not identical to the fragment shown above. Each differs from above by a single line of code. The lines containing the changes are the ones identified by arrows. Some of these programs will function exactly like the program shown above. Others will work differently or not at all.

For each program, indicate whether:

- it works the **SAME AS THE ORIGINAL** program,
- a **SYNTAX ERROR** will be detected as soon as you compile the program (explain what line causes the error and why),
- the program will run, but its behavior is **DIFFERENT FROM THE ORIGINAL**, possibly because an error occurs at run time. If there will be an error, identify the line on which it will occur and explain the nature of the error. If no error occurs, explain what **result** would be when the loop terminates (or why there is no termination!) if the code fragment were run with string **rna** described above.

Start your answer with one of the three **bold** phrases above, so that if your explanation is unclear we can still decipher your intent.

a) ```java
    start = rna.indexOf("AUG");  
    String result = "";          
    while (start >= 0) {
        int end = rna.indexOf("UAG", start);  
        result = result + rna.substring(start + 3, end) + "\n";  
        start = rna.indexOf("AUG", end + 3);
    }
```  

b) ```java
    int start = rna.indexOf("AUG");  
    String result = "";  
    while (start >= 0) {
        int end = rna.indexOf("UAG", start);  
        result = result + rna.substring(start, end) + "\n";  // <----------
        start = rna.indexOf("AUG", end + 3);
    }
```  

c) ```java
    int start = rna.indexOf("AUG");  
    String result = "";  
    while (start >= 0) {
        int end = rna.indexOf("UAG", start + 3);  // <---------------
        result = result + rna.substring(start + 3, end) + "\n";  
        start = rna.indexOf("AUG", end + 3);
    }
```  

d) ```java
    int start = rna.indexOf("AUG");  
    String result = "";  
    while (start >= 0) {
        int end = rna.indexOf("UAG", start);  
        result = result + rna.substring(start + 3, end) + "\n";  
        start = rna.indexOf("AUG", start);  // <---------------
    }
```  

**Question 3.**

Consider two computers, A and B that are connected by a wire that transmits at a rate of 10 Mbps (10 million bits/second). Suppose that A sends 1000 bits to B and the distance between A and B is 60Km.

(a) How long is it from when A transmits the first bit until B receives it, assuming that data travels at the speed of light (3 × 10^8 meters/second)? That is, what will be the propagation time for these bits?
(b) How long does it take A to put all 1000 bits onto the wire? That is, what will be the transmission time for these bits?

(c) How long is it from when A starts sending the bits until B has completely received it?

(d) How wide is a single bit being transmitted between these two computers? That is, how far can light travel in the time it takes to send a single bit?

Question 4.
Consider the signal represented by the following diagram:

(a) Assuming that signal is intended to transmit data encoded using on-off keying and that each of the intervals between the thin vertical lines represents the transmission of one bit, what sequence of binary digits does this signal represent? (Hint: Yes, the answer should be 21 bits long.)

(b) Now, continue to assume that on-off keying is being used and that the vertical lines represent bit times, suppose that the data has been grouped into 8-bit frames with one start bit preceding each frame. Under these assumptions, what data does the signal represent?

(c) Finally, suppose we drop the assumption about on-off keying and the assumption that the vertical lines represent bit times. Instead, assume that the Manchester encoding scheme is being used. In addition, continue to assume that the data is grouped into 8-bit frames and that each frame is preceded by a start bit (i.e., a one transmitted using Manchester encoding precedes the data). Under these assumptions, what data does the signal represent?