1. Understanding code

Draw a representation of what the computer's memory looks like at the end of each of these programs:

```java
public class Shortcut.Assignments {
    public static void main(String[] args) {
        int x = 1;
        int y = 3;
        int z = 5;
        x += y;
        y -= 2;
        z /= 2;
        x++; y--;
    }
}

x: starts at 1, changes to 4, then 5
y: starts at 3, changes to 1, then 0
z: starts at 5, changes to 2

public class Simple-While {
    public static void main(String[] args) {
        int x = 1;
        while(x < 10) {
            System.out.println("x = " + x);
            x++;
        }
    }
}

x: starts at 1, changes to 2, then 3, then 4, then 5, ..., then 9, then 10

class Infinite-While {
    public static void main(String[] args) {
        boolean b = true;
        while(b) {
            System.out.println("are we there yet?");
        }
    }
}(NOTE: x = 10 does NOT get printed by this program)
```
public class Complex-Update {
    public static void main(String[] args) {
        int x = 2;
        while(x < 1000) {
            System.out.println("x = " + x);
            x = x * x;
        }
    }
}

x: starts at 2, then 4, then 16, then 256, then something really big, much bigger than 1000 (specifically, 65536)

public class While-With-Nested-If {
    public static void main(String[] args) {
        int x = 1;
        while(x < 100) {
            System.out.println("x = " + x);
            if(x % 2 == 0) {
                x++;
            } else {
                x *= 2;
            }
        }
    }
}

x: starts at 1, then 2, then 3, then 6, then 7, then 14, then 15, then 30, then 62, then 63, then 126

screen: displays "x = " plus each one of the above values EXCEPT 126

public class Simple-For {
    public static void main(String[] args) {
        for(int i = 0; i < 10; i++) {
            System.out.println("i = " + i);
        }
    }
}
i: starts at 0, then 1, then 2, then 3, ... then 9, then 10

screen: displays "i = " plus each of the above values EXCEPT 10

class Infinite-For {
    public static void main(String[] args) {
        for (; true; ) {
            System.out.println("are we there yet?");
        }
    }
}

screen
are we there yet?
... (ad infinitum)

class Complex-Update-For {
    public static void main(String[] args) {
        for (int i = 2; i < 100; i = i * i) {
            System.out.println("i = " + i);
        }
    }
}

i: starts at 2, then 4, then 16, then 256, then 65536

screen: displays "i = " plus each of the above values EXCEPT 65536

2. Writing Java Programs with Loops
   a. Write a program that reads in an int from the keyboard, and stores the int in a variable x. Then write a loop that prints "Hello, world!" to the screen x times.

import java.util.Scanner;
class HelloXTimes {
    public static void main(String[] args) {
        Scanner keyboard = new Scanner(System.in);
        int x = keyboard.nextInt();
        for (int i = 0; i < x; i++) {
            System.out.println("Hello, world!");
        }
    }
}

b. Write a program that reads a int N from the keyboard, and prints out all of the squares between 1 and N^2. For instance, if the user enters 3 for N, the program should display this on the screen:
   1 * 1 = 1
   2 * 2 = 4
   3 * 3 = 9
import java.util.Scanner;
public class Squares {
    public static void main(String [] args) {
        Scanner keyboard = new Scanner(System.in);
        int N = keyboard.nextInt();
        int square = 0;
        for(int i=1; i<=N; i++) {
            square = i * i;
            System.out.println(i + " * " + i + " = " + square);
        }
    }
}