```
import java.awt.Point;
import java.util.LinkedList;

public class Huber_Brennan_HW5 {

    public static void main(String args[]) {
        // Problem 1
        System.out.println("Problem 1: " + yourName
());

        // Problem 2
        System.out.println("Problem 2: " + axb(10,
20, 30));

        // Problem 3
        Point p1 = new Point();
        p1.setLocation(0.0, 0.0);
        Point p2 = new Point();
        p2.setLocation(1.0, 1.0);

        System.out.println("Problem 3: " + distance
(p1, p2));

        // Problem 4
        LinkedList<String> slst = new LinkedList<St
ring>();
        slst.add("Na");
        slst.add("na");
        slst.add("na");
        slst.add("na");
        slst.add("na");
        slst.add("Batman");

        slst = purge(slst, "na");

        System.out.print("Problem 4: ");
        for(String s: slst) {
            System.out.print(s + " ");
        }

        slst.clear();
```

```
        // Problem 5
        slst.add("Hello");
        slst.add("0");
        slst.add("testing");
        slst.add("false");
        slst.add("check");


        System.out.print("\nProblem 5:\n\tList: ");

        for(String s: slst) {
            System.out.print(s + " ");
        }
        System.out.println("\n\tNum Trues: " + coun
tTrues(slst));
        slst.clear();

        // Problem 6
        LinkedList<Integer> ilst1 = new LinkedList<
Integer>();
        ilst1 = buildList(-5);
        System.out.print("Problem 6p1: ");

        for(int i: ilst1) {
            System.out.print(i + " ");
        }

        ilst1.clear();
        ilst1 = buildList(5);

        System.out.print("\nProblem 6p2: ");
        for(int i: ilst1) {
            System.out.print(i + " ");
        }

        // Problem 7
        LinkedList<Integer> ilst2 = new LinkedList<
Integer>();
        ilst1.clear();

        ilst1.add(1);
        ilst1.add(2);
```

```
        ilst1.add(3);

        ilst2.add(100);
        ilst2.add(10);
        ilst2.add(1);

        System.out.println("\nProblem 7: " + dotPro
duct(ilst1, ilst2));

        ilst1.clear();
        ilst2.clear();

        // Problem 8
        ilst1 = multiples(-5, 12);
        System.out.print("Problem 8: ");
        for(int i: ilst1) {
            System.out.print(i + " ");
        }
        ilst1.clear();

        // Problem 9
        ilst1.add(1);
        ilst1.add(2);
        ilst1.add(3);
        ilst1.add(4);

        slst.add("Foo");
        slst.add("bar");
        slst.add(" ");
        slst.add("Jones");

        System.out.println("\nProblem 9p1: plus: "
+ runCMD("plus", ilst1).get(0));
        System.out.println("Problem 9p2: times: " +
 runCMD("times", ilst1).get(0));
        System.out.println("Problem 9p3: append: "
+ runCMD("append", slst).get(0));
        System.out.println("Problem 9p4: asdfjkl: "
 + runCMD("asdfjkl;", ilst1));
        System.out.println("Problem 9p5: cdr: " + r
unCMD("cdr", ilst1));
```

```
        // Problem 10
        String str = "Brennan Huber Testing LOLCode
";
        System.out.println("Problem 10:\n\tOrigina
l: " + str + "\n\tFlipped: " + charFlip(str));

    }

    // Problem 1
    public static String yourName() {
        return "Brennan Huber";
    }

    // Problem 2
    public static int axb(int a,int x, int b) {
        return (a*x) + b;
    }

    // Problem 3
    public static double distance(Point p1, Point p
2) {
        return Math.sqrt(Math.pow((p2.getX() - p1.
getX()), 2) + Math.pow((p2.getY() - p1.getY()), 2)
);
    }

    // Problem 4
    public static LinkedList<String> purge(LinkedLi
st<String> lst, String match) {
        LinkedList<String> purged = new LinkedList<
String>();

        for(int i = 0; i < lst.size(); i++) {
            if(!lst.get(i).equals(match)) {
                purged.add(lst.get(i));
            }
        }

        return purged;
    }

    // Problem 5
```

```
    public static int countTrues(LinkedList<String>
 lst) {
        int trues = 0;

        for(int i = 0; i < lst.size(); i++) {
            if(lst.get(i).equals("0") || lst.get(i)
.equals("false")) {
            } else {
                trues++;
            }
        }

        return trues;
    }

    // Problem 6
    public static LinkedList<Integer> buildList(int
 number) {
        LinkedList<Integer> ll = new LinkedList<Int
eger>();

        if(number > 0) {
            for(int i = number; i > 0; i--) {
                ll.add(i);
            }
        } else {
            for(int i = number; i < 0; i++) {
                ll.add(i);
            }
        }

        return ll;
    }

    // wrong
    // Problem 7
    public static int dotProduct(LinkedList<Integer
> v1, LinkedList<Integer> v2) {
        int total = 0;

        if(v1.size() != v2.size()) {
            return -1;
```

```
        }

        for(int i = 0; i < v1.size(); i++) {
            total = total + (v1.get(i)*v2.get(i));
        }

        return total;
    }

    // Problem 8
    public static LinkedList<Integer> multiples(int
 base, int n) {
        LinkedList<Integer> buildlst = new LinkedLi
st<Integer>();
        LinkedList<Integer> multiples = new LinkedL
ist<Integer>();

        buildlst = buildList(n);

        for(int i: buildlst) {
            multiples.add(i*base);
        }

        return multiples;
    }

    // Problem 9
    public static LinkedList<?> runCMD(String opcod
e, LinkedList<?> lst) {

        if(opcode.equals("plus")) {
            LinkedList<Integer> ll = new LinkedList
<>();
            int total = 0;

            for(Object i: lst) {
                total = total + (int) i;
            }

            ll.add(total);

            return ll;
```

```
        } else if(opcode.equals("times")) {
            LinkedList<Integer> ll = new LinkedList
<>();
            int total = 1;

            for(Object i: lst) {
                total = total * (int) i;
            }

            ll.add(total);

            return ll;

        } else if(opcode.equals("append")) {
            String total = "";

            for(Object s: lst) {
                total = total + s;
            }

            LinkedList<String> ll = new LinkedList<
>();
            ll.add(total);

            return ll;

        } else if(opcode.equals("cdr")) {
            lst.remove();
        }

        return lst;
    }

    // Problem 10
    public static String charFlip(String s) {
        String total = " ";
        char c;

        for(int i = 0; i < s.length(); i++) {
            c = s.charAt(i);
            if(Character.isUpperCase(c)) {
```

```
                c = Character.toLowerCase(c);
            } else {
                c = Character.toUpperCase(c);
            }
            total = total + c;
        }

        return total;
    }
}
```