Brennan Huber
KHS616
HW5

To start, I originally was doing this very esoteric language I know, LOLCode. I provided the source code just so you can see that I wasn't trying to be lame and just do a basic language. I however had to abandon this project yesterday, due to how primative that language actually is. If LOLCode had a list structure or a better implemented array structure I gladly would've continued with that language, however it was more work than it was worth to create these myself. I hope you understand.

Java compared to scheme is incredibly different. The main issue that I encounted had to do with inputs into the methods. In scheme you had a list, and that list can hold many different types of data. But since Java is statically typed, you would have to have a list for ints, strings, ect. This mainly just circles back around to Java being a statically typed language. The main way I counted this (specifically in run-cmd) was to implement a generic as a parameter, that way I could just pass in any of the required parameters, and do the correct typed methods on them, as well as the return statement being the correct typ. For instance, if I hadn't done this then the append method would have caused an error if the list was full of ints.

Honestly, that is the only real problem I had during this assignment, Java is such an easy language that it has options to do just about anything you're assigned.

```
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));
```

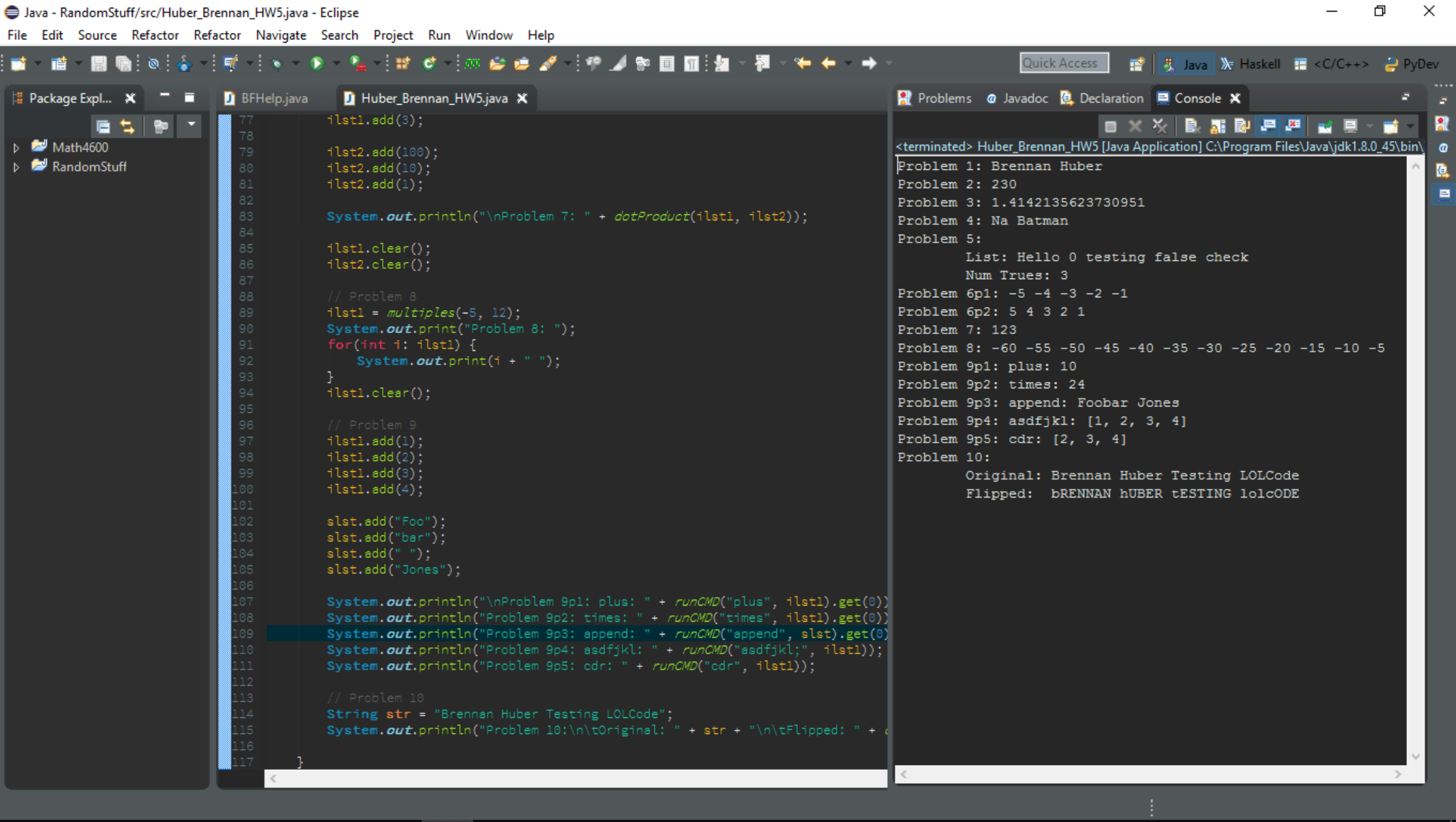
```
        // Problem 10
        String str = "Brennan Huber Testing LOLCode
";
        System.out.println("Problem 10:\n\tOrigina
l: " + str + "\n\tFlipped: " + charFlip(str));

    }

    // 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;
    }
}
```


Java - RandomStuff/src/Huber_Brennan_HW5.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Java Haskell <C/C++> PyDev
Package Expl...
Math4600
RandomStuff
BFHelp.java
Huber_Brennan_HW5.java
ilst1.add(3);
ilst2.add(100);
ilst2.add(10);
ilst2.add(1);
System.out.println("\nProblem 7: " + dotProduct(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: " + runCMD("cdr", ilst1));
// Problem 10
String str = "Brennan Huber Testing LOLCode";
System.out.println("Problem 10:\n\tOriginal: " + str + "\n\tFlipped: " +
}
Problems Javadoc Declaration Console
<terminated> Huber_Brennan_HW5 [Java Application] C:\Program Files\Java\jdk1.8.0_45\bin\
Problem 1: Brennan Huber
Problem 2: 230
Problem 3: 1.4142135623730951
Problem 4: Na Batman
Problem 5:
List: Hello 0 testing false check
Num Trues: 3
Problem 6p1: -5 -4 -3 -2 -1
Problem 6p2: 5 4 3 2 1
Problem 7: 123
Problem 8: -60 -55 -50 -45 -40 -35 -30 -25 -20 -15 -10 -5
Problem 9p1: plus: 10
Problem 9p2: times: 24
Problem 9p3: append: Foobar Jones
Problem 9p4: asdfjkl: [1, 2, 3, 4]
Problem 9p5: cdr: [2, 3, 4]
Problem 10:
Original: Brennan Huber Testing LOLCode
Flipped: bRENNAN hUBER tESTING lolcODE

```
BTW Brennan Huber
BTW HW5

HAI
   CAN I HAS STDIO?
   CAN I HAS STRING?

   BTW = Problem 1, DONE
   HOW DUZ I yourname
      FOUND YR "Brennan Huber:)"
   IF U SAY SO

   BTW = Problem 2, DONE
   BTW = looks like arguments are stored on a stack
..
   HOW DUZ I axb YR b AN YR x AN YR a

      I HAS A ax ITZ PRODUKT OF a AN x
      I HAS A total ITZ SUM OF ax AN b

    FOUND YR total
   IF U SAY SO

   BTW = Problem 3
   HOW DUZ I distance YR p1 AN YR p2
      I HAS A first ITZ 0
      I HAS A second ITZ findNextSpace p1 AN first

      I HAS A x1 ITZ getWord p1 first second
      first R second
      second R DIFF OF LEN OF p1 AN 1
      I HAS A y1 ITZ getWord p1 first second

      second R findNextSpace p2 AN first

      I HAS A x2 ITZ getWord p1 first second
      first R second
      second R DIFF OF LEN OF p2 AN 1
      I HAS A y2 ITZ getWord p2 first second

      x1 IS NOW A NUMBAR
      y1 IS NOW A NUMBAR
```

```
      x2 IS NOW A NUMBAR
      y2 IS NOW A NUMBAR

      BTW do the distance formula.

      I HAS A lh
      I HAS A rh

      lh R DIFF OF x2 AN x1
      lh R PRODUKT OF lh AN lh

      rh R DIFF OFF y2 AN y1
      rh R PRODUKT OF rh AN rh

      I HAVE A ans

      ans R SUM OF lh AN rh
      ans R sqrt ans

      FOUND YR ans

   IF U SAY SO

   BTW since there is no freakin library have to wr
ite everything by hand.
   HOW DUZ I sqrt YR x
      I HAS A xn ITZ 10
      I HAS A lim ITZ 100
      I HAS A i ITZ 0

      IM IN YR LOOP UPPIN YR i WILE i SMALLR THAN l
im
         I HAS A ans ITZ QUOSHUNT OF x AN xn

         ans R SUM OF sn AN ans
         ans R QUOSHUNT OF ans AN 2

         xn R ans
      IM OUTTA YR LOOP

      FOUND YR xn
   IF U SAY SO
```

```
   BTW = Problem 4,
   HOW DUZ I purge YR match AN YR lst

   IF U SAY SO

   BTW = Problem 5,
   HOW DUZ I countTrues YR lst
     I HAS A trues ITZ 0

     I HAS A first ITZ -1
     I HAS A second ITZ 0
     I HAS A i ITZ 0
     I HAS A word ITZ ""
     I HAS A var

     second R findNextSpace lst first

     IM IN YR LOOP UPPIN YR i TIL BOTH SAEM i AN LE
N OF lst
        BTW create a word
        word R getWord lst first second
        VISIBLE word

        BTW check if it is a literal 0
        BOTH SAEM word AN "0", O RLY?
           NO WAI
              trues R SUM OF trues AN 1
        OIC

        BTW cast as a troof
        var R word
        var IS NOW A TROOF

        BTW if it is true increment trues
        var, O RLY?
           YA RLY
              trues R SUM OF trues AN 1
        OIC

        BTW reset word to empty string
```

```
        word R ""

        BTW calculate new first and second
        first R second
        second R findNextSpace lst first
      IM OUTTA YR LOOP

      FOUND YR trues
    IF U SAY SO

    BTW = Problem 6,
    HOW DUZ I buildlist YR n

    IF U SAY SO

    BTW = Problem 7,
    HOW DUZ I dotproduct YR v1 AN YR v2

    IF U SAY SO

    BTW = Problem 8,
    HOW DUZ I multiples YR base AN YR N

    IF U SAY SO

    BTW = Problem 9,
    HOW DUZ I runcmd YR opname AN YR lst
       I HAS A i ITZ 0
       I HAS A length, BTW size of the list
       I HAS A var ITZ 0

       opcode, WTF?
          OMG "plus"
             IM IN YR LOOP UPPIN i TIL BOTH SAEM i A
N length
                var R SUM OF var AN lst!i  BTW var =
 var + lst[i]
             IM OUTTA YR LOOP

             FOUND YR var
          OMG "times"
             var R 1
```

```
            IM IN YR LOOP UPPIN i TIL BOTH SAEM i A
N length
               var R PRODUKT OF var AN lst!i  BTW v
ar = var * lst[i]
            IM OUTTA YR LOOP

            FOUND YR var
         OMG "append"
            I HAS A str ITZ ""
            IM IN YR LOOP UPPIN i TIL BOTH SAEM i A
N length
               SMOOSH str AN lst!i  MKAY BTW append
 list[i] onto the string.
            IM OUTTA YR LOOP
         OMG "cdr"
            BTW do scheme cdr here
         OMGWTF
            FOUND YR FAIL
      OIC

   IF U SAY SO

   BTW = Problem 10,
   HOW DUZ I charflip YR str
      BTW === main part of this function =======
      I HAS A i ITZ 0
      I HAS A new_str ITZ ""
      IM IN YR LOOP UPPIN i TIL BOTH SAEM i AN LEN
OF str
         isLower str!i, O RLY?
            YA RLY
               SMOOSH new_str AN up str!i MKAY
            NO WAI
               SMOOSH new_str AN low str!i MKAY
         OIC
      IM OUTTA YR LOOP

      FOUND YR new_str

   IF U SAY SO

   BTW charflip helper: if it is lower case, up it
```

```
   HOW DUZ I up YR char
      I HAS A i ITZ 0
      I HAS A upper "QWERTYUIOPASDFGHJKLZXCVBNM"
      I HAS A lower "qwertyuiopasdfghjklzxcvbnm"

      IM IN YR LOOP YR i TILL BOTH SAEM i AN LEN OF
 upper
         BOTH SAEM lower!i AN char, O RLY?
            YA RLY
               FOUND YR upper!i
         OIC
      IM OUTTA YR LOOP

   IF U SAY SO

   BTW charflip helper: if it is upper case, lower
it.
   HOW DUZ I low YR char
      I HAS A i ITZ 0
      I HAS A upper "QWERTYUIOPASDFGHJKLZXCVBNM"
      I HAS A lower "qwertyuiopasdfghjklzxcvbnm"

      IM IN YR LOOP YR i TILL BOTH SAEM i AN LEN OF
 upper
         BOTH SAEM upper!i AN char, O RLY?
            YA RLY
               FOUND YR lower!i
         OIC
      IM OUTTA YR LOOP

   IF U SAY SO

   BTW charflip helper: test to see if the characte
r is lower case.
   HOW DUZ I isLower YR char
      I HAS A i ITZ 0

      IM IN YR LOOP UPPIN YR i TILL BOTH SAEM i AN
LEN OF upper
         BOTH SAEM lower!i AN character, O RLY?
            YA RLY
               FOUND YR WIN
```

```
         OIC
      IM OUTTA YR LOOP

      FOUND YR FAIL

   IF U SAY SO

   BTW: creates a word from the two spaces given.
   HOW DUZ I getWord YR lst YR first YR second
      I HAS A i ITZ SUM OF first AN 1
      I HAS A word ITZ ""

      IM IN YR LOOP UPPIN YR i TIL BOTH SAEM i AN s
econd
         word R SMOOSH word AN lst!i MKAY
      IM OUTTA YR LOOP

      FOUND YR word
    IF U SAY SO

   BTW takes the current space, and returns the nex
t space, or the length of the string.
   HOW DUZ I findNextSpace YR lst YR second
      I HAS A i ITZ SUM OF second AN 1

      IM IN YR LOOP UPPIN YR i TILL BOTH SAEM i AN
LEN OF lst
         BOTH SAEM lst!i AN " ", O RLY?
            YA RLY
               FOUND YR i
         OIC
      IM OUTTA YR LOOP

      FOUND YR LEN OF lst
    IF U SAY SO


   BTW ========================= main ============
=====

   VISIBLE yourname
   VISIBLE axb 5 6 1
```

```
    VISIBLE
    VISIBLE
    VISIBLE
    VISIBLE
    VISIBLE
    VISIBLE
    VISIBLE
    VISIBLE charflip "Brennan Huber tESTING LOLcode"

KTHXBYE
```