Sample problem for #4

{

blog = linda.universe\_rd(("MicroBlog", linda.TupleSpace))[1]

blog.\_out(("bob","distsys","I am studying chap 2"))

blog.\_out(("bob","distsys","The linda example's pretty simple"))

blog.\_out(("bob","gtcn","Cool book!"))

blog = linda.universe\_rd(("MicroBlog", linda.TupleSpace))[1]

blog.\_out(("alice","gtcn","This graph theory stuff is not easy"))

blog.\_out(("alice","distsys","I like systems more than graphs"))

blog = linda.universe.\_rd(("MicroBlog",linda.TupleSpace))[1]

#Chuck's search statements

t1 = blog.\_rd(("bob","distsys",str))

t2 = blog.\_rd(("alice","gtcn",str))

t3 = blog.\_rd(("bob","gtcn",str))

#random search statement

t = blog.\_rd((str,"distsys",str))

}

Answers:

{

A) Making the above statement find the next tuple, rather than a random one.

Extend the tuple statements by adding a fourth parameter to act as an index #. I want to be able to iterate through each index. Pseudo code below:

{

int i = 1;

blog = linda.universe\_rd(("MicroBlog", linda.TupleSpace))[1]

blog.\_out((i = 1, "bob","distsys","I am studying chap 2"))

blog.\_out((i = 2, "bob","distsys","The linda example's pretty simple"))

blog.\_out((i = 3, "bob","gtcn","Cool book!"))

blog = linda.universe\_rd(("MicroBlog", linda.TupleSpace))[1]

blog.\_out((i = 4, "alice","gtcn","This graph theory stuff is not easy"))

blog.\_out((i = 5, "alice","distsys","I like systems more than graphs"))

while result.topic == "distsys":

i = i + 1

t = blog\_rd((i, str, "distsys", str))

}

B) By adding a layer of security and requiring a form of authentication before posting as a certain name should prevent poster name thievery.

}