

**Does the order of the rules matter (in other words, would it matter if we rearranged the order of the cond clauses)? Explain why or why not.**

To illustrate why order does matter, I picked out a few sample conversations. This first conversation demonstrates that if we add a word to a conversation, it will change the case that ends up printing the response.

*You: First I would like to get to know you*  
*Shrink: first you would like to get to know me*  
*You: first I would like to love you*  
*Shrink: All is fair in love and war.*

After we run the "You-Me" filter, both conversations have the word "you" in them (The "I" is changed to "you"). As a result, the first response is presented by the case that looks for the word "you." The second conversation has the word "love" in it, which is picked up before the "you" case. As a result, we get the "love" response.

The next conversation illustrates the same idea, except we use the word "bye."

*You: maybe tomorrow*  
*Shrink: Be more decisive!*  
*You: maybe tomorrow. goodbye.*  
*Shrink: Goodbye.*

In general, if we add any word that has the substring "bye", it will automatically give the response "Goodbye." This response is triggered by the first case, so having the substring "bye" takes precedence over all other cases.

This last case illustrates how the order may prevent us from reaching one of the cases.

*You: Do you think I should quit?*  
*Shrink: Why do you want to think?*

Since the first line has the word "think" in it, which is a verb, the first case that is met has the condition involving the verbp function. This function simply determines if the input string has a verb in it and returns it. Let's note that there is another case afterwards that looks for the string "Do you think" (or "Do me think" after we run the "you-me" filter). This case is much more specific than the case involving the verbp function. The verbp function is much more broad since it only looks for the string "think." As a result, we will never actually reach the "do you think" case unless we switch the two cases.

In my implementation I went ahead and switched the cases around so now we get the following conversation:

*You: Do you think I should quit?*

*Shrink: I think you should answer that yourself.*

*You: I think I should quit*

*Shrink: Why do you want to think?*

### **Is the program convincing as a psychological advisor, or conversationalist? Why or why not?**

The program is able to carry a conversation. It either repeats a lot of what the user has already said or it uses a generic response. Regardless, the idea is to get more information from the person and encourages the user to continue to speak. The program simply leads the user into talking more.

As far as being a convincing psychological advisor, the program seems to encourage you to ask yourself questions about your problem. Granted the questions are generic, they do seem to give the user more to think about with regards to their problem.

### **How would you extend the program so that had more intelligence?**

The first thing I would add is the ability to handle greetings better. The sample conversation illustrates the program's current ability:

*You: Hi, My name is nick*

*Shrink: hi, your name is nick*

This response seems to be mocking the user. (I know my own name!) The ability to recognize words such as “hi” and “hello” and respond with a simple “hello” would suffice.

The next thing I would add is the ability to use past tense. When conversing with someone, improper usage sticks out. For example:

*You: When I was a child I had a fever*

*Shrink: when you was a child you had a fever*

Instead this should read “When you were a child....”

Although it would be quite the feat, the ability to use proper grammar would give the program more natural sounding responses. For example, this responses doesn't sound right at all:

*You: I hope you have a good night*

*Shrink: you hope me have a good night*

Differentiating between “I” and “me” is just one of those things in the English language that requires looking at context to determine proper usage. In general, the ability to respond with respect to context would add a lot in terms of intelligence to the program.