**transformIWantStatement()**

private String transformIWantToStatement(String statement)

{

statement = statement.trim();

String lastChar = statement.substring(statement.length() -1);

if(lastChar.equals("."))

{

statement = statement.substring(0, statement.length()-1);

}

int psn = findKeyword(statement, "I want to",0);

String restOfStatement = statement.substring(psn + 9);

return "What would it mean to" + restOfStatement + "?";

}

When the user input is “I want to \_\_\_”, the MagPie responds with “What would it mean to \_\_\_\_\_” automatically.

SIMILAR: My code is almost identical to the answer. The only difference is some of the formatting of the code, but that doesn’t change anything(the answer just has some of the lines tabbed in). Also, another difference is that I initiated restOfStatement to equal statement.substring(psn+9) when the provided answer says psn+6. This doesn’t change anything though. Also, the provided answer has a .trim() in the restOfStatement variable. However, when I put this .trim() in my code, it made the response crunched where “What would it mean to eat” became “What would it mean toeat.”

EFFICIENCY: My code is more efficient thanthe answer because they both do the same thing: with a user input of “I want to \_\_\_”, the computer responds with “What would it mean to \_\_\_\_\_”, but my code doesn’t squeeze the words together.

WHAT COULD YOU HAVE DONE DIFFERENTLY? I could have used the indexOf() method but we were supposed to use the findKeyword() method since it produces little to no errors. My code is efficient though so there’s no need to change it.

WHAT DID YOU LEARN FROM COMPARING YOUR CODE TO THE PROVIDED ANSWER?

I learned that my code and the provided answer are so similar. This means that I am finally understanding how to use the findKeyword() method. I also learned that the .trim() method takes out spaces.

**transformYouMeStatement()**

private String transformYouMeStatement(String statement)

{

statement = statement.trim();

String lastChar = statement.substring(statement.length() -1);

if(lastChar.equals("."))

{

statement = statement.substring(0, statement.length()-1);

}

int psnOfYou = findKeyword(statement, "you",0);

int psnOfMe = findKeyword(statement, "me", psnOfYou + 3);

String restOfStatement = statement.substring(psnOfYou + 3, psnOfMe);

return "What makes you think that I" + restOfStatement + "you?";

}

When the user input is “You \_\_\_\_ me”, the MagPie responds with “What makes you think that I \_\_\_\_\_ you” automatically.

SIMILAR: My code is almost identical to the answer. The only difference is some of the formatting of the code, but that doesn’t change anything(the answer just has some of the lines tabbed in). Also, the provided answer has a .trim() in the restOfStatement variable. However, when I put this .trim() in my code, it made the response crunched where “What makes you think that I love you” became “What makes you think that I loveyou.”

EFFICIENCY: My code is more efficient than the answer because they both do the same thing: with a user input of “You \_\_\_\_\_ me”, the computer responds with “What makes you think that I \_\_\_\_\_ you” automatically, but my code doesn’t squeeze the words together.

WHAT COULD YOU HAVE DONE DIFFERENTLY? I could have used the indexOf() method but we were supposed to use the findKeyword() method since it produces little to no errors. My code is efficient though so there’s no need to change it.

WHAT DID YOU LEARN FROM COMPARING YOUR CODE TO THE PROVIDED ANSWER?

I learned that my code and the provided answer are so similar. This means that I am finally understanding how to use the findKeyword() method. I also learned that the .trim() method takes out spaces.