

```

import java.util.Scanner;

/**
 * MagpieRunner3
 * @author YOUR NAME AND CLASS PERIOD
 */
public class MagpieRunner3
{

    public static void main(String[] args)
    {
        // *****
        // ***** StringExplorer part of Magpie Activity 3 *****
        // *****

        String sample = "The quick brown fox jumped over the lazy dog.";

        // Demonstrate the indexOf method.
        int position = sample.indexOf("quick");
        System.out.println("sample.indexOf(\"quick\") = " + position);

        // Demonstrate the toLowerCase method.
        String lowerCase = sample.toLowerCase();
        System.out.println("sample.toLowerCase() = " + lowerCase);
        System.out.println("After toLowerCase(), sample = " + sample);

        // Using indexOf when the desired substring is not found

        // Using indexOf to find multiple occurrences of the desired substring

        // *****
        // ***** END OF StringExplorer part of Magpie Activity 3 *****
        // *****

        System.out.println("\n\n\n\n"); // blank lines in output to separate parts of program

        Magpie3 maggie = new Magpie3();

        System.out.println(maggie.getGreeting());
        Scanner in = new Scanner (System.in);
        String statement = in.nextLine();

        while (!statement.equals("Bye"))
        {
            System.out.println(maggie.getResponse(statement));
            statement = in.nextLine();
        }
    }
}

```

```
}// end of MagpieRunner3
```

```
class Magpie3
```

```
{  
    /**  
     * Get a default greeting  
     *  
     * @return a greeting  
     */  
    public String getGreeting()  
    {  
        return "Hello, let's talk.";  
    }  
  
    /**  
     * Gives a response to a user statement  
     *  
     * @param statement  
     *         the user statement  
     * @return  
     *         a response based on the rules given  
     */  
    public String getResponse(String statement)  
    {  
        String response = "";  
  
        if (statement.length() == 0)  
        {  
            response = "Say something, please.";  
        }  
        else if (findKeyword(statement, "no") >= 0)  
        {  
            response = "Why so negative?";  
        }  
        else if (findKeyword(statement, "mother") >= 0  
                || findKeyword(statement, "father") >= 0  
                || findKeyword(statement, "sister") >= 0  
                || findKeyword(statement, "brother") >= 0)  
        {  
            response = "Tell me more about your family.";  
        }  
        else  
        {  
            response = getRandomResponse();  
        }  
  
        return response;  
    }  
  
    /**  
     * Search for one word in phrase. The search is not case
```

```

* sensitive. This method will check that the given goal
* is not a substring of a longer string (so, for
* example, "I know" does not contain "no").
*
* @param statement
*         the string to search
* @param goal
*         the string to search for
* @param startPos
*         the character of the string to begin the
*         search at
* @return
*         the index of the first occurrence of goal in
*         statement or -1 if it's not found
*/
private int findKeyword(String statement, String goal, int startPos)
{
    String phrase = statement.trim().toLowerCase();
    goal = goal.toLowerCase();

    // The only change to incorporate the startPos is in the line below
    int psn = phrase.indexOf(goal, startPos);

    // Refinement--make sure the goal isn't part of a word
    while (psn >= 0)
    {
        // Find the string of length 1 before and after the word
        String before = " ";
        String after = " ";

        if (psn > 0)
        {
            before = phrase.substring(psn - 1, psn);
        }

        if (psn + goal.length() < phrase.length())
        {
            after = phrase.substring(psn + goal.length(), psn + goal.length() + 1);
        }

        // If before and after aren't letters, we've found the word

        // before is not a letter
        if (((before.compareTo("a") < 0) ||
            (before.compareTo("z") > 0)) && ((after.compareTo("a") < 0) ||
            (after.compareTo("z") > 0)))
        {
            return psn;
        }

        // The last position didn't work, so let's find the next, if there is one.

```

```

        psn = phrase.indexOf(goal, psn + 1);

    }

    return -1;
} // end of findKeyword(String, String, int) method

/**
 * Search for one word in phrase. The search is not case
 * sensitive. This method will check that the given goal
 * is not a substring of a longer string (so, for
 * example, "I know" does not contain "no"). The search
 * begins at the beginning of the string.
 *
 * @param statement
 *         the string to search
 * @param goal
 *         the string to search for
 * @return the index of the first occurrence of goal in
 *         statement or -1 if it's not found
 */
private int findKeyword(String statement, String goal)
{
    return findKeyword(statement, goal, 0);
} // end of findKeyword(String, String) method

/**
 * Pick a default response to use if nothing else fits.
 *
 * @return a non-committal string
 */
private String getRandomResponse()
{
    final int NUMBER_OF_RESPONSES = 4;
    double r = Math.random();
    int whichResponse = (int) (r * NUMBER_OF_RESPONSES);
    String response = "";

    if (whichResponse == 0)
    {
        response = "Interesting, tell me more.";
    }
    else if (whichResponse == 1)
    {
        response = "Hmmm.";
    }
    else if (whichResponse == 2)
    {
        response = "Do you really think so?";
    }
    else if (whichResponse == 3)

```

```
{
    response = "You don't say.";
}

    return response;
} // end of getRandomResponse method

} // end of Magpie3
```