1.(Comparing Strings) Write an application that uses String method compareTo to compare two strings input by the user. Output whether the first string is less than, equal to or greater than the second.

2.(Comparing Portions of Strings) Write an application that uses String method regionMatches to compare two strings input by the user. The application should input the number of characters to be compared and the starting index of the comparison. The application should state whether the strings are equal. Ignore the case of the characters when performing the comparison.

3. (Random Sentences) Write an application that uses random-number generation to create sentences. Use four arrays of strings called article, noun, verb and preposition. Create a sentence by selecting a word at random from each array in the following order: article, noun, verb, preposition, article and noun. As each word is picked, concatenate it to the previous words in the sentence.
The words should be separated by spaces. When the final sentence is output, it should start with a capital letter and end with a period. The application should generate and display 20 sentences.
The article array should contain the articles **"the", "a", "one", "some" and "any"**;
 the noun array should contain the nouns **"boy", "girl", "dog", "town" and "car"**;
 the verb array should contain the verbs **"drove", "jumped", "ran", "walked" and "skipped"**;
the preposition array should contain the prepositions **"to", "from", "over", "under" and "on"**.

4. (Tokenizing Telephone Numbers) Write an application that inputs a telephone number as a string in the form **(555) 555-5555**. The application should use method split to extract the area code as a token, the first three digits of the phone number as a token and the last four digits of the phone number as a token. The seven digits of the phone number should be concatenated into one string. Both the area code and the phone number should be printed. Remember that you'll have to change delimiter characters during the tokenization process.

5.(Displaying a Sentence with Its Words Reversed) Write an application that inputs a line of text, tokenizes the line with String method split and outputs the tokens in reverse order. Use space characters as delimiters.

6. *(Displaying Strings in Uppercase and Lowercase)* Write an application that inputs a line of text and outputs the text twice—once in all uppercase letters and once in all lowercase letters.

7. *(Tokenizing and Comparing Strings)* Write an application that reads a line of text, tokenizes the line using space characters as delimiters and outputs only those words beginning with the letter "b".

8. *(Tokenizing and Comparing Strings)* Write an application that reads a line of text, tokenizes it using space characters as delimiters and outputs only those words ending with the letters "ED".

9. Write your own versions of String search methods indexOf and lastIndexOf.

10. *(Creating Three-Letter Strings from a Five-Letter Word)* Write an application that reads a five-letter word from the user and produces every possible three-letter string that can be derived from the letters of that word. For example, the three-letter words produced from the word "bathe" include "ate," "bat," "bet," "tab," "hat," "the" and "tea."

11. *(Printing Dates in Various Formats)* Dates are printed in several common formats. Two of the more common formats are **04/25/1955** and **April 25, 1955**
Write an application that reads a date in the first format and prints it in the second format.

12. Accept a website name as a command line argument & display its constituent parts.

    i.e. www.yahoo.com

        www

        yahoo

        com