Pulling Strings

Rules:

- You ARE allowed to use char, String
- Utilize "loops" and "if" conditions as necessary
- Use Scanner class when taking input
- Use of SimpleInput is NOT allowed anymore
- This lab is to test your EXPERTISE in using the String class, so show maximum usages

Take help from:
http://www.javabat.com
http://java.sun.com/docs/books/tutorial/java/data/strings.html
See items containing the term "**String**" under "**2. Data Type**" category from
http://www.java2s.com/Tutorial/Java/CatalogJava.htm
ftp://192.168.0.84/javadoc/api/java/lang/String.html
http://java.sun.com/javase/7/docs/api/java/lang/String.html
books?

Task 1

Input a word into a String. Print the length of the word, that is, how many characters / symbols / spaces are there. Take help from:
ONLY First 4 lines of "**String Length**" paragraph from
http://download.oracle.com/javase/tutorial/java/data/strings.html
http://download.oracle.com/javase/7/docs/api/java/lang/String.html#length%28%29

Sample input 1:
Programming
Sample output 1:
11

Sample input 2:
hunger
Sample output 2:
6

Task 2

Input a word into a String. Print each character on a line by itself. Use charAt method. Take help from:
http://www.homeandlearn.co.uk/java/charAt.html
http://www.java2s.com/Code/Java/Language-Basics/StrCharAtshowStringcharAt.htm

Sample input 1:
Programming
Sample output 1:
P

r
o
g
r
a
m
m
i
n
g

Sample input 2:
hunger
Sample output 2:
h
u
n
g
e
r

Task 3

There are two ways to print ASCII code / Unicode / int value of each character.
String s = "Bye";
Either,

1) char ch = s.charAt(0);
   int codeForCh = (int) ch;
   System.out.println(codeForCh); // this line would print **66** which is the code for **character B**

Or,

2) int codeForCh = s.codePointAt(0);
   System.out.println(codeForCh); // this line would print **66** which is the code for the **character B**

Now your task is to input a word into a String. Then print code for each character in the String using the 2nd method discussed above. Use any means necessary.
To check if your program is working correctly or not, you can find a list of all correct values from the following website. Look at "**Dec**" and "**Char**" columns only, ignore **Hex** column.
http://www.cdrummond.qc.ca/cegep/informat/Professeurs/Alain/files/ascii.htm

Sample input 1:
Programming
Sample output 1:
P : 80
r : 114
o : 111

g : 103
r : 114
a : 97
m : 109
m : 109
i : 105
n : 110
g : 103

Sample input 2:
hunger
Sample output 2:
```
h : 104
u : 117
n : 110
g : 103
e : 101
r : 114
```

Task 4

Imagine user will give you some numbers, say 8 numbers. All numbers are between 2 and 4. One program to count these numbers may be:

```
Scanner sc=new Scanner(System.in);
int[] statistics=new int[5];
int i, n;
System.out.println("enter total eight numbers between 2 and 4 only, otherwise I will have to make
my array larger.");
for(i=0;i<=7;++i){
	n=sc.nextInt();
	statistics[n]=statistics[n]+1;
}
for(i=2;i<=4;++i){
	System.out.println(i+" was found "+ statistics[i]+ " times");
}
/*
If the user enters 2, 4, 3, 4, 4, 2, 2, 2. The program above will print:
2 was found 4 times
3 was found 1 times
4 was found 3 times
*/
Also note that we have left index 0 and index 1 unused as we do not need those. We only need to
count numbers between 2 and 4.
```

If you looked closely at the website mentioned in Task3, you must have realized that all codes are between 0 and 255. Taking help of the example shown above, print the statistics of occurrence of each character on a line by itself. Assume that user will only give CAPITAL letters. So you will have to count values of CAPITAL letters only.

Sample input 1:
BANGLADESH

SAMPLE output 1:
```
A which is 65 was found 2 times
B which is 66 was found 1 times
```

```
C which is 67 was found 0 times
D which is 68 was found 1 times
E which is 69 was found 1 times
F which is 70 was found 0 times
G which is 71 was found 1 times
H which is 72 was found 1 times
I which is 73 was found 0 times
J which is 74 was found 0 times
K which is 75 was found 0 times
L which is 76 was found 1 times
M which is 77 was found 0 times
N which is 78 was found 1 times
O which is 79 was found 0 times
P which is 80 was found 0 times
Q which is 81 was found 0 times
R which is 82 was found 0 times
S which is 83 was found 1 times
T which is 84 was found 0 times
U which is 85 was found 0 times
V which is 86 was found 0 times
W which is 87 was found 0 times
X which is 88 was found 0 times
Y which is 89 was found 0 times
Z which is 90 was found 0 times
```

Task 5

Input a word into a String.
Print the word.
Print the word again after adding "==THE END==" at the end of the word.
Then print the word again.
Your whole program may contain the word "String" at most two times.
You are NOT allowed to use concat method.
You ARE allowed use '+' operator to concatenate (join?) words.

Take help from:
http://www.java2novice.com/java_string_examples/concat/
http://www.codingdiary.com/developers/developers/diary/javaapi/java/lang/SampleCode/ConcatStringExampleCode.html
http://www.java-samples.com/showtutorial.php?tutorialid=217
http://www.tutorialspoint.com/java/java_string_concat.htm
"Concatenating Strings" paragraph from http://download.oracle.com/javase/tutorial/java/data/strings.html

Sample input 1:
Programming
Sample output 1:
Programming
Programming==THE END==
Programming

Sample input 2:
hunger
Sample output 2:
hunger
hunger==THE END==
hunger

Task 6

Input a word into a String.
Print the word.
Print the word again after adding "==THE END==" at the end of the word.
Then print the word again.
Your whole program may contain the word "String" at most two times.
You HAVE TO to use concat method.
You are NOT allowed use '+' operator to concatenate (join?) words.
Take help from:
http://www.java2novice.com/java_string_examples/concat/
http://www.codingdiary.com/developers/developers/diary/javaapi/java/lang/SampleCode/ConcatStringExampleCode.html
http://www.java-samples.com/showtutorial.php?tutorialid=217
http://www.tutorialspoint.com/java/java_string_concat.htm
"**Concatenating Strings**" paragraph from http://download.oracle.com/javase/tutorial/java/data/strings.html

Sample input 1:
Programming
Sample output 1:
Programming
Programming==THE END==
Programming

Sample input 2:
hunger
Sample output 2:
hunger
hunger==THE END==
hunger

Task 7
Show examples of the following methods of the String class. Try to cover as several possible usages/example for each of these methods.
7a) CompareTo(s1.compareTo(s2) means 1st word-2nd word in ascii)
7b) compareToIgnoreCase
7c) startsWith(s2.startsWith(s1) means both word will be start same.(t/f))
7d) endsWith
7e) equals//return the Boolean of word

```
7f)  equalsIgnoreCase
7g)  indexOf
7h)  lastIndexOf
7i)  length
7j)  replace(char oldChar, char newChar)
7k)  substring( minues the letter)
7l)  toCharArray
7m)  toLowerCase(convert into small letters)
7n)  toUpperCase((convert into capital letters)
7o)  trim//remove the space
7p)  valueOf//change the format+print value
////-1=means nothing happend
```