CS Lab 100

**Single Loops and Arrays**

# Part 1:

**Filename**: DontLeaveMe.java

Collect a String from the user, and repeat back whatever it says. The program should continue to do this until the user enters the String “Exit”. Ignore capitalization in “Exit”. You *may not* use a break or System.exit statement. The loop should naturally end when the user types “Exit”. Here is a sample run:

What should I say? *Hello, World!*  
Hello, World!

What should I say? *I rule the world with my iron first! And my hot-pink wand!*  
I rule the world with my iron first! And my hot-pink wand!

What should I say? *exit*  
Exiting...

# Part 2:

**Filename**: GetE.java

The irrational number e, also called Euler’s number, is approximately 2.71828. Euler proved that this number was irrational by showing that it was equal to the infinitely expanding series 1 + 1/1 + 1/(2!) + 1/(3!) + 1/(4!)…

getE should take a command-line argument of a single number n. It should then perform the number of calculations from the list above specified by n, and output the result to the screen. So,

> java getE 1

1.0

> java getE 2

2.0

> java getE 3

2.5

> java getE 4

2.666666667

> java getE 5

2.708333333

**HINT**: use Integer.parseInt(args[0]) to turn the first String argument passed in into an int.

**NOTE**: While there is an obvious solution to this problem that involves a nested loop, the best solution involves only a *single* loop. See if you can figure out the single-loop solution!

# Part 3:

**Filename**: SpreadLines.java

SpreadLines will take a series of arguments at the command line and print them, one to a line. For instance:

> java SpreadLines Hello there discombobulated friend

Hello

there

discombobulated

friend

> java SpreadLines

> java SpreadLines 34

34

# Part 4:

**Filename**: Scrambler.java

Scrambler will contain the method scramble(int addBy, int numTimes, String scrambleThis)

It will then return a scrambled version of the String. The scrambled version will have every addByth word, rotating around the String numTimes times. So:

scramble(1,3,"Hello my joyful friend! I missed you yesterday!");

Will return "Hello my joyful".

Alternatively, scramble(2,4,"Hello my joyful friend! I missed you yesterday!");

Will return "Hello joyful I you"

scramble(2,12,"Hello my joyful friend! I missed you!");

Will return "Hello joyful I you! my friend! missed Hello joyful I you! my"

In order to do this, take scrambleThis and use the split method.

scrambleThis.split(" ") will return an array of Strings with the space characters as the cutting points. You can then use loops and the array to accomplish your task.

# Part 5:

**Filename**: EncryptionV2.java

The goal here is to take your work from the previous encryption lab and extend it to be really useful. We’re going to allow it to take all of the standard characters from a space to a tilde, including all of the standard punctuation and capital and lowercase letters. We’re also going to allow arbitrarily long Strings.

Here’s what you need to know:

The standard range of characters goes from 32 (a space) to 126 (a tilde), giving us 95 possible characters.

You can use your basic logic from before, but you will now need a loop to go through your String. The methods are:

public String encrypt(int key, String text)

public String decrypt(int key, String text)

*Here are some values you can use to test your methods:*

encrypt (7, "a") should become h.

encrypt (1, "Hi there, friend!") should return

ISs)28KQ}>EXbhw|>

so decrypt (1, "ISs)28KQ}>EXbhw|>") should return

Hi there, friend!

Encrypt with 16 and the String:

Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.

Should get you:

Vf|0Pdhx,2RTch)=CZ`o0JPRey:<DTt%;Nnuw-6<Oc$':J`hq'GN^q'0P`o0ENXl-1APeo~%4Iu68Xgm&FUWlv'6b#'7FJPZqw|=GVvcmpv\*?Y&FHW\|"(-7;=RX]}3Ccx"(HYl|.>R\q{,;[py{1QS`m.<BQqs'-MQdjl"(-MSe{}+Y

(Please note that both the original text and the resulting encrypted text are on 1 line. Any wrapping is a result of line endings and Word.)