# ITEC2150 - Test 2 - Coding

# Recursion

Create a program that returns the number of double letters in a word using recursion.

For example, the word “aaron” has two double letters and returns 1. The word “carrr” returns 2 because there are two pairs including the overlapping pair. These are some of the test cases that your method should pass:

“” 🡪 0

“aa” 🡪 1

“ab” 🡪 0

“aaa” 🡪 2

“aabb” 🡪 2

“aaabb” 🡪 3

“aaaabbb” 🡪 5

Please turn in the file “Recursion.java” which contains the method doubleCounter(String input).

# File I/O and Generics

~~Create a program that reads in the text file, input.txt. Your program should write the file to another text file in reversing the letters in each word. The reversing method should be a bound generic which accepts an ArrayList<String>. For example, if the input file is:~~

~~This is an input an input line.~~

~~This is the second input line.~~

~~Your program should write:~~

~~sihT si na tupni na tupni .enil~~

~~sihT si eht dnoces tupni .enil~~

~~to the file. The file writing should be handled in a non-bound generic method.~~

~~To test your program, you should be able to use your output file as the input to get the original file.~~

~~Submit MirrorFile.java.~~

# LinkedList

~~Create a java program that allows the user to enter as many names and birthdays as they like. Each of these should be stored in an object of the supplied Birthday class. You must create an InvalidBirthdayException that will be used by Birthday. Your drivers should handle the exceptions thrown. Once the names are entered, you should print the list using the Birthday toString() method.~~ Finally, create an iterator to traverse the list and return the oldest person in the list.

Additional instructions/hints.

1. ~~The birthday class contains two instance variables, a String name and an int bday. Defaults will be Fred with a birthday of 20160714.~~
2. ~~Birthdays should be entered in the format yyyymmdd.~~
3. ~~You should structure your program with methods, recommend one to enter the values for the list, one to print the list,~~ and one to find the oldest.
4. findOldest must use an iterator to traverse the list, you are not allowed to sort on the birthday to find the oldest. printList can use an enhanced for to print the contents of the list.
5. ~~You need to modify Birthday to implement Comparable. This should be used to find the oldest person. Think about who would be the oldest~~

Example output

Enter the person's name. Enter -1 to exit

Name cannot be empty.

Enter the person's name. Enter -1 to exit

Sally

Enter the birthday (yyyymmdd)

0

Enter the person's name. Enter -1 to exit

Fred

Enter the birthday (yyyymmdd)

-19961212

Birthday of -19961212 is not valid. Must be between 0 and 20160714

Enter the birthday (yyyymmdd)

19961212

Enter the person's name. Enter -1 to exit

Sue

Enter the birthday (yyyymmdd)

20160715

Birthday of 20160715 is not valid. Must be between 0 and 20160714

Enter the birthday (yyyymmdd)

2016071$

Invalid characters found in the entered birthday

Enter the birthday (yyyymmdd)

20160714

Enter the person's name. Enter -1 to exit

-1

Sally 0

Fred 19961212

Sue 20160714

The oldest person in the list is Sally 0

Submit Birthday.java and BirthdayTester.java.