COMP1039 – PGP - Java
Lab 3

3.1

Define a left rotation method on an array. Given an integer d, shift each element in the array d steps left and then return the result. This method takes an integer d, and an array of strings as its input (from the command-line), it then performs the left rotation and output the result array to the console window.

For example, if the input is 3 one two three four, then the output should be:

four one two three

3.2

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward.

Given a string A from the command line, return yes if A is a palindrome.

For example,

mam	Yes
ob	No

3.3

Write a program that creates a “triangular” two-dimensional array **ts** of 10 rows. The first row has length 1, the second row has length 2, the third row has length 3 and so on. Then initialize the array using nested for loops so that the value of **ts[i][j] is i+j**. Finally, print out the array in a nice triangular form.

3.4

Given a time in 12-hour AM/PM format, convert it to military (24-hour) time. Remember, 12:00:00AM is 00:00:00 on a 24-hour clock, and 12:00:00PM is 12:00:00 on a 24-hour clock.

Input Format

hh:mm:ssAM or hh:mm:ssPM

Output Format

hh:mm:ss

3.5

Given an array of integer, write a merge sort to sort the elements from smallest to largest.

3.6

You are given a number of sticks of varying lengths. You will iteratively cut the sticks into smaller sticks. Every time you cut the sticks based on the shortest one until there is none left. For example, if you have an array of sticks {2, 3, 5, 5, 2}, the first time 2 will be cut from all these sticks which results in a new array {1, 3, 3}. We then cut 1 from each stick and have our new array {2, 2}. After the next round, no sticks left.

Given an array of integer representing the length of each stick, write a program which output the number of remaining sticks in each round.

3.7

You are starting at index 0 of an n-element array game. For some index i, only the following moves are valid:

1) you can move from game[i] to game[i+1] if the value of game[i+1] equals to 0.

2) you can also move backwards from game[i] to game[i-1] if the value of game[i-1] equals to 0 (also i-1 must be greater than or equals to 0)

3) you can also jump from game[i] to game[i+leap] if the value of game[i+leap] equals to 0.

When you reach the end of the array or current i+leap is greater than or equals to the length of the array you win the game.

Given a game array and a leap value, return "Yes" if there is a way to win the game, return "No" otherwise.

For example, given 0 0 0 1 1 1 and 5, Return Yes

Given 0 1 0 and 1 return No