**WRAV202: Practical 1**

# Instructions

This practical must be completed before your practical session of the week of 1 to 5 August 2022. A student assistant will check your solution and assign marks during the practical session.

You also have to submit your code on the WRAV202 learn.mandela.ac.za page. Please copy all your code into a single .cs file. You will only be able to upload one file per task.

This practical must be your own, original work. You are encouraged to ask questions from the student assistants and may discuss potential solutions with your fellow students BUT, you may not copy any code from another student or from the internet.

## Task 1

The files *File1.txt* and *File2.txt* each contains a list of words. Some words appear in both files and some words only appear in one of the files. The files are sorted. Consider the following class definition for a linked list node that stores a string:

class SLLNode

{

public String cargo;

public SLLNode next;

}

You are reminded that class String contains the following method:

public int CompareTo (String strB)

This method returns a number less than zero if the current String instance precedes strB, zero if the instance is equal to strB and a number greater than zero if the instance follows strB.

Do the following:

1. Write a method that will open both files and then uses an appropriate merge algorithm to obtain a linked list that contains all the words that occur in **both** files. Your method should not store the contents of the files in memory, the linked list should be created as the files are read. You may only make one pass over each of the files and one pass over the linked list (i.e. your code should run in linear time). The head pointer to the linked list should be returned by your method.
2. Write a recursive method to reverse a linked list (given a head pointer as input). Use the following method declaration:

public SLLNode Reverse(SLLNode curhead)

//Preconditions: curhead is a reference to the first element in a list (or sub-list)

//Post conditions: a reference to the first element of the reversed list is returned

1. Use the methods you wrote to merge the contents of the files and display the result in reverse order.

## Task 2

Consider the following class definition for a doubly linked list node that stores an integer:

class DLLNode

{

public int cargo;

public DLLNode next;

public DLLNode prev;

}

1. Write a method that will insert a value in the middle index (rounded down) in the doubly linked list. For example, were you to successively insert the numbers 1,2,3,4,5,6 into the middle of the list, your the list would contain the numbers in the following order after each insert:

1

2 1

2 3 1

2 4 3 1

2 4 5 3 1

2 4 6 5 3 1

You will have to maintain a head pointer, a tail pointer and a count of the number of elements in the list. Use your method to successively insert the first 15 integers into the middle of the list.

1. Write a method to efficiently display the list in reverse order.

## Task 3

You decide to give your little sister 12 “presents” for her birthday, one of each of the coins and notes currently in circulation in South Africa:

5c, 10c, 20c, 50c, R1, R2, R5, R10, R20, R50, R100, R200

After a few days she mentions that she still has 8 of her presents left unspent. Write a **recursive** program to determine all the possibilities of the amounts that your sister could still have left. For example, part of the output of your program should look something like this:

10c, R1, R2, R10, R20, R50, R100, R200, = R383.1

10c, R1, R5, R10, R20, R50, R100, R200, = R386.1

10c, R2, R5, R10, R20, R50, R100, R200, = R387.1

20c, 50c, R1, R2, R5, R10, R20, R50, = R88.7

20c, 50c, R1, R2, R5, R10, R20, R100, = R138.7

20c, 50c, R1, R2, R5, R10, R20, R200, = R238.7