**WRAV202: Practical 7**

Write a separate program for each of the following questions. Each program must read an input file and write to an output file.

# Question 1 – Merge sequences

This question will involve merging two sequences of numbers in order. Each line of the input file contains sequences of numbers. Your program must process two lines at a time. Below is an example of the input file.

10,18,22,100

45,23,8,5,2,1

20,34,38,79

102,45,39,22

9,14,35

89,68,34,20

For every two lines, the first line is an ascending sequence, and the second line is a descending sequence. You must write a program that uses a stack and a queue to combine every two lines into a single ascending sequence. Put the first line into a queue, and the second line into a stack. Then compare the first value in the queue to the top value of the stack to determine which is the next value that must be written to the output. Write each output sequence to its own line, with a comma trailing each number. For the above example, the *Outputs.txt* file is:

1,2,5,8,10,18,22,23,45,100,

20,22,34,38,39,45,79,102,

9,14,20,34,35,68,89,

# Question 2 – Roulette

Roulette is a game of chance available in most casinos. A ball lands on either a red or a black cavity on a spinning wheel. Gamblers bet money on which colour the ball will land. If they guess the correct colour, they double their money, otherwise they lose their money.

You heard from an informant that the roulette table at a casino has been set up so that the ball lands in a predictable sequence of red or black. There is still some randomness (to fool most gamblers), but you can determine the sequence by analysing enough sequences that occurred. You record red-black sequences that you observed in a text file using ‘R’ for red and ‘B’ for black, for example:

BRR

R

BRB

BBR

BRB

BR

RRB

BRB

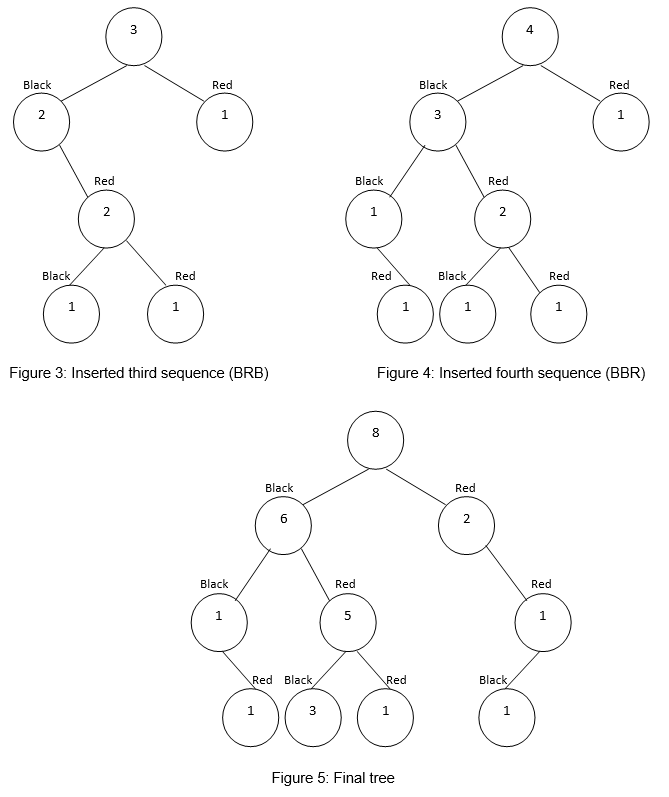
Each observed sequence can be of any length. You can construct a tree using these sequences. Each node should record the number of occurrences of a colour, given the sequence of previous colours. For example, the first sequence will yield the tree shown in Figure 1.

Diagram

Description automatically generated

Inserting the second sequence will alter the tree to the tree in Figure 2.

The third sequence will change the tree into the tree shown in Figure 3 while the fourth sequence will alter the tree to the tree shown in Figure 4. Finally, after inserting all sequences, the tree will appear as depicted in Figure 5.



The root node records the total number of responses. From Figure 5 it can be seen that the most likely sequence is: BRB, since the values in the nodes are the largest (6, 5 and 3).

Write code to create a binary tree based on the red-black sequences in *Inputs.txt*. There will always be an even number of sequences. Nodes should only be created in the tree if a corresponding sequence exists. Once the tree has been created, transverse the tree and output the most likely sequence of colours which has occurred in at least half of the sequences. Write the output to a file called *Outputs.txt*. Should two colours be equally likely, assume that it will be red. In the example above, there are 8 sequences, so we output the most likely sequence of colours which has occurred 4 or more times. Our most likely sequence is BRB, and of that sequence, only the first two colours occur 4 or more times. So, the *Outputs.txt* file will contain:

BR

.

# SUBMISSION

You have to submit two *.cs* files for this practical.