Imperial College of Science, Technology and Medicine	University of London
Computer Science (CS) / Software Engineering (SE)	BEng and MEng Examinations Part I
Department of Computing	Integrated Laboratory Course

Laboratory work is a continuously assessed part of the examinations
and is a required part of the degree assessment.
Laboratory work must be handed in for marking by the due date.
Late submissions may not be marked.

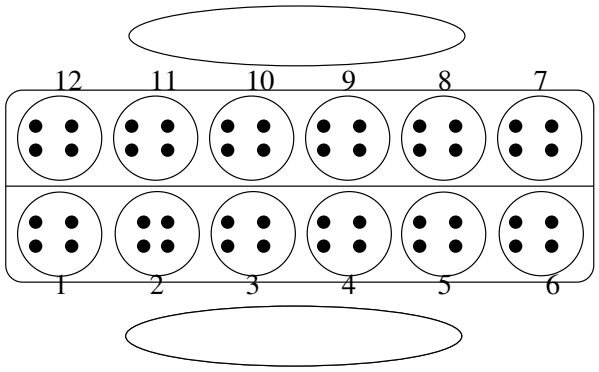
Exercise: 9	Working: Individual
Title: Owari	
Issue date: 8th December 2003	Due date: 15th December 2003
System: Linux	Language: Kenya/Java

Aims

- To design and write a simple game-playing program in Kenya/Java.

The Problem

- Write a Kenya/Java program **Owari.k**/ **Owari.java** to manage a game of Owari between two human players.

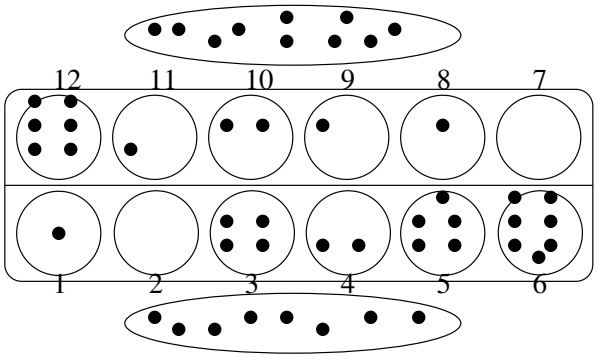


The Owari board at the start of a game.

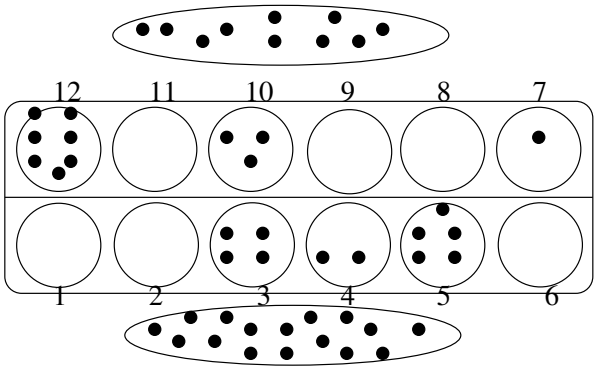
Figure 1

The Game of Owari

- Owari is a game widely played throughout West Africa. There are no standard rules for Owari and the rules vary between the different places where it is played. The program that you are asked to write should allow two players to play Owari according to the following rules:
- Owari is played on a board containing 12 bowls divided into two rows of six bowls. At the start of the game each bowl contains 4 stones (see Figure 1).



Player 1 selects bowl 6 for their move



Player 1 captures 8 stones

Figure 2

- The two players take it in turn to make moves. One of the players is selected to make the first move.
- The first player can only move stones from the bottom row and the second player can only move stones from the top row.
- A move consists of choosing a bowl, removing all the stones from it and then, moving in an anti-clockwise direction, placing one stone in each bowl following, until all the

stones have been placed. Note that the board wraps around with the "last" bowl for one player being next to the "first" bowl for the other player.

- If a player has no stones on his/her side of the board then that player must pass his/her move. On the other hand a player must move if a move is possible.
- If a bowl contains a single stone before a move is made and ends up with two or more stones after the move, then the player captures (removes) all the stones in that bowl. The removed stones are credited to the mover's score. (An illustration of a move is given in Figure 2.)
- The game is over when one of the players scores a total of at least 24 stones. There are some rare situations where the game continues forever but for the purposes of your program you can assume that all games will end with a clear winner.
- If you do use different rules for your Owari game, please explain them to your PPT.

Submit by Monday 15th December 2003

What To Do

- Copy the skeleton program **Owari.k** into your working directory by typing **exercise 9**.
- Try out the lab version, which is called **LabOwari**, so that you can get some practice playing the game. Notice that the bowls are labelled 1 .. 12 in an anti-clockwise direction starting from the bowl in the bottom left hand corner extending to the bowl in the top left corner. Each bowl has 4 stones in it initially. There are are also two bowls, one under and one on top of the board which are used for keeping captured stones. It should be fairly obvious from the user interface how you interact with the program. Note particularly any error messages that are displayed.
- Note: the lab version is written in a different language and window system and is provided to allow you to try some practice Owari games.
- The skeleton program provides some types and I/O methods such as, **getMove**, **displayScoreAndBoard** and **putMove** along with the method for initialising a board **initialiseBoard**. We have also provided a type for the player's' scores and the board. Initially you should use these types and methods but as the exercise is going to be marked by demonstration you can change them if you wish. You can also write the program in java if you want to.
- The main program should manage a single game of Owari. To start a game the various data structures used will have to be initialised and the initial state of the game displayed. After this the following steps will be repeated until the game is finished. The next player who can move is selected and a move is received from the player at the keyboard. A message should be displayed if a player has to pass a move or if a player attempts to input an invalid move. A move will be invalid if either it is made from the wrong side of the board or if there are no stones in the bowl chosen. Once the move is validated the board and players' scores are updated. The updated board and score is then displayed.

Unassessed

Getting the Computer to Play Against You

- Try extending your program so that, in addition to managing a game and keeping the score, it generates moves for one of the players with the object of winning the game.
- To allow a program to play Owari, against either a human player or another program, the program must be able to take the role of either player.
- A program which takes the role of player 1 should generate the first move. It will then wait for the contestant to type in the next move which will have been generated by the opposing player or program which has taken the role of player 2. The program will then alternately generate moves and get moves from the keyboard until the game is over. If two programs are playing each other, moves generated by one program are typed into the opposing program and vice versa.
- After a move is generated by the program it should be displayed using the method **putMove**. The program should then display the board and update the score.
- You will need to devise and implement a strategy to generate a move given the state of the board at any time.
- To deal with the cases where it is impossible to capture all the stones you may wish to count the number of moves that have been made since the last capture and declare the game over after a fixed number of moves were made without a capture.
-

If you write this unassessed game playing version, you can enter the **2003 Owari Tournament**. There will be a prize of a £25 book token, plus some book tokens donated by O'Reilly publishers. Further details of the tournament are given on a separate sheet, handed out either with this exercise spec. or very soon after. May the best program win.

Submission

- Submit in the usual way by typing **submit 9**.

Assessment

- This exercise will be marked by demonstration to your PPT. You should arrange a time for the demonstration in the second last or last week of term.

```
Main program loop (plays one game)          1
Code for getting and validating move        1.5
Code for making move on board               2.5
```

Design, style, readability	5
Total	10