

UNIVERSITY OF DUBLIN

TRINITY COLLEGE

Faculty of Engineering, Mathematics and Science
School of Computer Science and Statistics

B.A. (Mod.) Computer Science
Senior Freshman Examination

Trinity Term 2008

2BA3 - Systems Programming

Thursday 5 June 2008 Goldhall 09:30 - 12:30

Dr. Kevin Casey

Attempt any **FIVE** questions
All questions are worth 20 marks

All programs should be commented, indented and divided into functions and classes/abstract data types (ADTs) appropriately.

Exam candidates may use a non-programmable calculator

1. (a) Distinguish between `auto`, `static` and suitable scenarios where each would mean for a function to be static in C?
When might you want to use the `register` keyword?
What is the downside to the using this keyword?
- (b) Write `egrep` regular expressions for the
- (i) A string ending in `a` and beginning
 - (ii) A 5-letter palindrome
 - (iii) An email address
 - (iv) A line containing only 3 identical I

2. Noughts and Crosses is a game played in a 3x3 grid, where one player 'X' plays against a second player 'O'. Player X always starts first and places an X anywhere in the 3x3 grid. Then player O puts an 'O' in any free space on the grid. This process continues until either:
- (i) The board fills up with X's and O's (a draw)
 - (ii) A player manages to get three X's or three O's in a line (horizontally, diagonally or vertically). In this case the player owning the three symbols is the winner.

Design and write a C function which accepts a 3x3 array of characters representing the state of a Noughts and Crosses board. The function should then return one of the following values:

- (i) '?' if the board is not valid. For example a board with two X's and no O's.
- (ii) '#' if the board is valid, but the game has not finished
- (iii) '=' if the game has finished and the board is full (a draw)
- (iv) 'X' if the game has finished and X has won
- (v) 'O' if the game has finished and O has won.

The array should only contain the following characters (and your function should check this):

- 'X' - this square belongs to player X
- 'O' - this square belongs to player O
- ' ' - (a space) this square is empty.

(Total: 20 marks)

3. Bit stuffing is a technique used in data transmission, where a certain bit sequence (in this case 01111110) has a special meaning – the stop flag. This presents a problem if the data a user wants to transmit contains this sequence, since it might inadvertently be interpreted as a stop flag, instead of actual data.

The solution is to use a technique called 'bit-stuffing'. Before data is transmitted, the data is scanned from left to right. Any time a series of five binary 1's is encountered, a binary 0 is inserted into the data immediately afterwards. This prevents the stop flag (01111110) occurring in the user data.

When the binary data is received, the original data must be restored. The technique here is to scan the data from left to right for any sequence of five binary 1's and to drop the next digit (which would have to be a binary 0).

Write a C++ class, `StuffBits`, using the list container from the STL to store the binary digits. You should provide for the following methods/operators:

```
stuff()
destuff()
<< (Stream insertion operator)
```

You should use an iterator for both the `stuff` and `destuff` methods.

(Total: 20 marks)

4. Write a C++ class to represent complex numbers. Your class should provide +, − and * operators (no division operator is necessary). You should also overload the << and >> operators for input and output. Write a small template method that takes two numbers as parameters, and writes the sum of the numbers to the screen. The method should work with both integers and complex numbers as the template type.

(Total: 20 marks)

5. (a) Write a C++ class to represent a stack of integers. Ensure you supply an appropriately commented constructor and destructor. Your class should throw exceptions in the case of underflow or overflow.
- (b) Using only the Stack class you have just written for storage purposes, write a C++ class to represent a queue of integers. Ensure you supply an appropriately commented constructor and destructor. Your class should throw exceptions in the case of underflow or overflow.

(10 marks)

Hint: Each Queue will require two Stacks, and either the enqueue or dequeue operation will quite likely be inefficient.

(10 marks)

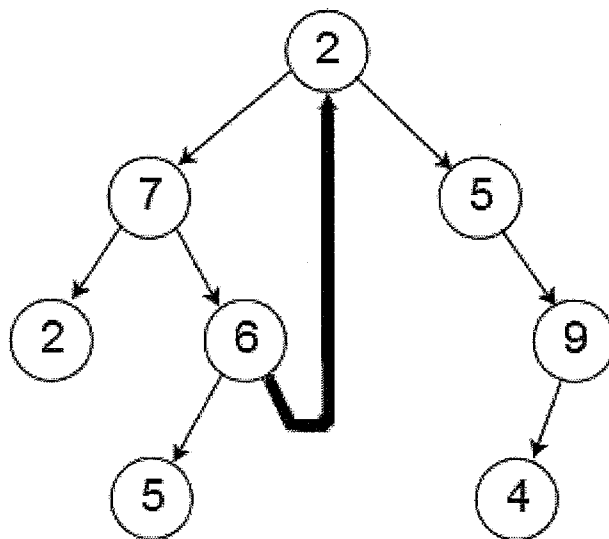
(Total: 20 marks)

6. One of your co-workers is developing code that involves a binary tree written in C, built from nodes defined as follows:

```
struct binary_tree_node
{
    int data;
    struct binary_tree_node * left;
    struct binary_tree_node * right;
};
```

Unfortunately your colleague has a problem. They suspect that one of the nodes has a broken pointer and instead of pointing to a child, is pointing to ancestor (see following diagram for an example).

Given a pointer to the root node of a binary tree, write a C function that lists the address of all nodes that are broken in this way. You may assume that if any node points to one of its own ancestors, then it is



broken.

Figure 1: A broken binary tree. The node containing 6 is the broken node.

The faulty pointer is highlighted in bold.

(Total: 20 marks)

7. (a) What is the standard template library (STL)? Briefly describe the most important container classes in the STL. With many class libraries of container data structures, it is necessary to have a separate version of the code for each combination of data structure, type of contained object, and algorithm that can be applied. Explain how the STL avoids this sort of code duplication using templates and iterators.

(12 marks)

- (b) Design and write a C++ class called PhoneBook to represent a phone book. The numbers and names should be stored as a map. Apart from a suitable constructor, your class should provide for the following methods:

`void add(string name, string number)` – add the name, number entry to the PhoneBook. If there is already an entry for the name, the old number will be overwritten

`void listAll()` - display a list of all entries, sorted by name

`string findNumber(string name)` – find the number associated with this name

(8 marks)

(Total: 20 marks)