UNIVERSITY OF DUBLIN

TRINITY COLLEGE

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

B.A.(Mod.) Computer Science **Trinity Term 2009**
Senior Freshman Examination

Systems Programming (2BA3)

Thursday 4th June 2009 Goldsmith Hall 09:30 – 12:30

Dr David Gregg

Instructions to Candidates:

□ Answer 4 out of the 6 questions
□ All questions are marked out of 25

Materials permitted for this examination:

□ Calculator.

1. The most commonly encountered logics use two states: true and false. However, ternary (or three-valued) logics, where values are one of true, false, or unknown can also be useful. Write a C++ class to represent ternary values. Your class should provide && (and), || (or) and ! (not) operators. You should also overload the << and >> operators for input and output. A truth table for the ternary logic operators appears below.

A	B	A || B	A && B	!A
True	True	True	True	False
True	Unknown	True	Unknown	False
True	False	True	False	False
Unknown	True	True	Unknown	Unknown
Unknown	Unknown	Unknown	Unknown	Unknown
Unknown	False	Unknown	False	Unknown
False	True	True	False	True
False	Unknown	Unknown	False	True
False	False	False	False	True

[25 marks]

2. You are running the phone-in voting service for a major international song contest, and you need to make sure that only one vote is registered from each phone number in the country. It is important that you can decide quickly either register the vote or reject it due to a previous call from the same number. Your boss is so worried about detecting duplicate votes quickly that she has told you that your part of the program must, on average, be able to register or reject a vote in constant time.

Write C abstract data type that keeps track of all phone numbers seen so far, and returns whether or not a phone number has been seen already. For completeness, you should also have functions for removing a phone number from the set, and for freeing any allocated memory when the set of phone numbers is no longer needed. You may assume that phone numbers are null-terminated C strings.

[25 marks]

3. Write a C++ string tokenizer class, where a token is defined as a maximum sequence of non-whitespace characters in a string. Your class should support the following methods:
   - A constructor that initializes the class with a C++ string to be tokenized.
   - A constructor that initializes the class with a C-style null-terminated string to be tokenized.
   - A method that returns a string representing the next token in the string that is being tokenized.
   - A method that returns true if there are no more tokens in the string, and false otherwise.

   You should not use any tokenizing functions or classes from the C or C++ standard libraries.

   [25 marks]

4. Many applications allocate very large numbers of blocks of memory of the same size. For example, a program that uses a lot of linked lists may allocate large numbers of list nodes. The C functions *malloc* and *free* are designed to deal with memory blocks of any size, and as a result of this generality they incur significant time and space overheads.

   One solution to this problem is to use a memory pool. For example, a memory pool for linked lists of integers might allocate a single piece of memory large enough for 256 linked list nodes using a single call to malloc. These list nodes are then added to a free list within the memory pool abstract data type (ADT). When memory is requested from the ADT, it returns an item from the free list. If the free list is empty, then the ADT must allocate another large block of memory for more list nodes. When a list node is no longer needed, it is returned to the free list. When the memory pool is no longer needed, each of the large blocks of memory must be freed. Write a C ADT representing a memory pool. The ADT should support the following functions:

```
// create a new memory pool with memory for size list nodes
mempool* mempool_new(int size);

// allocate a list node from the memory pool
listnode * mempool_allocate(mempool* pool);

// return a list node to the memory pool
void mempool_free(mempool * pool, listnode* item);

// destructor; free all memory used by memory pool
void listpool_delete(mempool * pool);
```

[25 marks]

5. 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.

[15 marks]

To show the flexibility of the STL, write a C++ template function that computes the average of a set of numbers stored in an STL container. Your function should be capable of dealing with any of the main STL containers (except map and multimap) and should be able to accept any numeric type that supports the + and / operators. To show your function in action, give a few examples of how the code can be called with actual types.

[10 marks]

6. Write a C++ function to compute the mass of a molecule, given its chemical formula. Your function should have the following prototype:
int compute_mass(String formula, map<String, int> weights);

The first parameter is a C++ string containing the chemical formula of the compound. For example, H_2SO_4 would be represented by the string "H2SO4". Similarly, NaCl would be represented with the string "NaCl".

The second parameter is an STL map, mapping strings to integers. For each element, this map records its corresponding atomic weight. Note that the first letter in the name of an element is always upper case, and subsequent letters are always lower case. Therefore, it is always possible to know whether the next character is part of the name of the current element (lower case) or the start of the name of a new element (upper case).

[25 marks]