UNIVERSITY OF DUBLIN

TRINITY COLLEGE

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

B.A.(Mod.) Computer Science **Trinity Term 2010**
Senior Freshman Examination
B.A.(Mod.) MSISS
Junior Sophister Examination

Systems Programming I and II (CS2014/5)

Wednesday 5 May **RDS - Main** **14:00 – 17:00**

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:

Non-programmable calculators are permitted for this examination – please indicate the make and model of your calculator on each answer book used.

1. Most modern processors have special support for operating on short *vectors* of numbers. For example, the Intel SSE instructions have a special type representing a group of four 32-bit integers. A single machine instruction can operate on all four values at once. For example, if we represent such as vector as four values [a, b, c, d] then:
   [2, 6, 9, 4] + [3, 8, 7, 6] = [5, 14, 16, 10]
   Write a C++ class that describes a short vector of four integers. Your class should support +, -, * and / integer operators as well as the comparison operators ==, > and <. The comparison operators should return an integer with one bit for each of the four values. The bit should be either 1, indicating the result is true, or 0, indicating false. For example:
   [2, 6, 9, 4] > [3, 8, 7, 6] = $0010_2 = 2_{10}$
   Your short vector class should also support << and >> operators for writing to and reading from standard library streams.

   [25 marks]

2. One way to represent sets of items is to use a linked list. Write a C++ (not C) class to represent sets of items. Your class should be a template class, so that the type of items to be stored is specified at point where variables of the class are declared. Your class should provide functions to add, remove, and check for membership of the set, as well as functions to implement set union and set intersection. You should not use the Standard Template Library in your solution.

   [25 marks]

3. The C programming language allows characters, integers and other types to be written to files, but it provides no direct support for writing information to a file in chunks of less than one byte. This is unfortunate because many applications can save file space by saving streams of individual bits, or irregular sized chunks of data.

Write a C (not C++) abstract data type (ADT) of a file that allows information to be written to the file one bit at a time. Your ADT should support operations to open a new file, write a bit to the file, and close the file. Note that you will need to use the normal C file features within your ADT. C does not support writing single bits to a file, so you will have to buffer the bits in a byte in memory until the buffer is full, and then write the full byte to the file. You also need to consider the situation where the file ends at an uneven bit number. Hint: The C function fputc(unsigned char c, FILE * file) is probably the most convenient for writing single bytes to a file.

[25 marks]

4. Write a C function that identifies whether or not a string is a valid email address. For the purposes of this question (the real rules are much more complex) an email address may be defined as follows. An email address consists of one or more identifiers separated by full stops, followed by the @ symbol, followed by one or more identifiers separated by full stops, followed by a full stop, followed by a terminator, followed by the end of the string. The function should have the following prototype:

   int isValidAddress(string email_address, char ** terminators, int num_terminators);

   A terminator is a final string, such as "com", "net" or "ie" that ends an email address. The set of all valid terminators is stored in the array *terminators*. The length of the terminators array is stored in the parameter *num_terminators*. An identifier is a sequence of one or more alphanumeric characters. You may use the functions isalpha(char) and isdigit(char) to identify alphabetic and numeric characters respectively.

5. The C++ Standard Template Library (STL) provides a set of standard container classes for use in C++ programs. One of the most important of these is the vector class. The following shows the broad outline of a simplified version of the STL vector class.

```
template <class T>
class myvector{
private:
        // add your own private variables and methods here
public:
        myvector();             // create new, empty vector
        myvector(int max);  // create new vector with max spaces
        ~myvector();            // destructor
        void push_back(T item);    // add new item onto end of list
        T pop_back();           // remove last space and return    contents
        T& at(int index);               // indexing with bound check
        T& operator[ ](int index);   // indexing without bound checking
};
```

Add the remaining declarations to this class, and provide the bodies of methods to implement each of the methods listed above. You may use the basic building blocks of the C++ language (arrays, classes) to construct your class, but you may not use the STL in your code.

[25 marks]

6. Write a C abstract data type (ADT) to represent a stack of characters. Using this ADT, write a program that reads in simple boolean expressions in postfix notation, evaluates the expression and writes out the result. The input will consist only of the characters 'T' (representing true) and 'F' (representing false) and the operators &&, || and !, representing logical and, or and not respectively. For example, if the input is:

   T T && F || !

   the output would be: F

   You can assume that there will always be a space between each Boolean value and/or operator, and that the input is always a valid postfix expression. You should read in the expression as a sequence of command line parameters to your program. [25 marks]