Coláiste na Tríonóide, Baile Átha Cliath
Trinity College Dublin
Ollscoil Átha Cliath | The University of Dublin

Faculty of Engineering, Mathematics and Science

School of Computer Science and Statistics

Integrated Computer Science — Trinity Term 2017
BA (Mod) Computer Science and Business
Year 2 Annual Examinations

Systems Programming I

Saturday 6th May 2017 — RDS Main Hall — 09.30-11.30

Dr David Gregg

Instructions to Candidates:

- Answer 2 out of the 3 questions
- All questions are marked out of 50
- All program code should be commented, indented and use good programming style

Materials permitted for this examination:

- Calculator

1. 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

    Boolean expressions in postfix notation are evaluated in exactly the same way as integer expressions in postfix. 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.

    [50 marks]

2. The most common representation of character strings is as an array of characters, but it is also possible to represent strings using linked lists. When strings are represented with a linked list, each character of the string occupies one node of the linked list.

   Write a C abstract data type (ADT) called mystring that represents a string as a linked list of characters. Your ADT should have functions to do the following:

   a) Create a new empty string [8 marks]
   b) Take a normal null-terminated array C string as a parameter, and returns a new linked list-based string of type mystring. [10 marks]
   c) A string compare function that takes two mystring parameters and returns -1 if the first string is less than the second, 0 if the two string are equal, and 1 if the first is greater than the second. [16 marks]
   d) A search function that takes two mystrings as parameters: a piece of text, and a string to search for. The function should search the text string for the search string, and if it finds a sequence of characters within the text that is equal to the characters in the search string it should return 1. Otherwise the function should return 0. [16 marks]

   You may not use any standard C string manipulation functions in your solution.

3. Modern computers support standard data sizes, such as 8-bit, 16-bit or 32-bit numbers. However, for some applications it is useful to be able to store arrays of data that are not a standard size. For example, one might want to store an array of 9-bit numbers.

   One way to work with arrays of 9-bit numbers is to store them in a packed format, but provide functions to unpack the data before operating on them. In a packed format, each 9-bit number is stored as nine consecutive bits in memory, and the next 9-bit number starts at the next bit following on. In other words, unlike standard data sizes, the elements of packed arrays of 9-bit numbers do not evenly line up along byte boundaries.

   Before operating on packed arrays of numbers we normally need to unpack them to an array of a standard data type that supports standard C arithmetic. For example, we might unpack a packed array of 7-bit unsigned integers into a standard unpacked array of 32-bit unsigned integers.

   Write a C function that takes as a parameter a packed array of unsigned integers and returns an array with the same values represented as standard unsigned integers. The function should look like:
   unsigned * unpack(unsigned char * data, int num_of_elements, int bits_per_element);
   Where data points to a packed array of elements, where bits_per_element is the number of bits in each element of the array, and where num_of_elements is the number of elements in the array. [25 marks]

   Write a second C function that takes as a parameter an array of unsigned integers and returns a packed array containing the same values but packed into the smallest number of bits. The function should look like:
   unsigned char * pack(unsigned * data, int num_of_elements, int bits_per_element);
   Where data points to an array of unsigned integers that should be packed, where bits_per_element is the number of bits from each element that should be put into the packed array, and where num_of_elements is the number of elements in the array.

   [25 marks]