UNIVERSITY OF DUBLIN

TRINITY COLLEGE

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

Integrated Computer Science Programme **Trinity Term 2014**

B.A. (Mod.) Business and Computing

Systems Programming I (CS2014)

Saturday, 10th May 2014 **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. One of the most commonly-used data structures in programming is the linked list. The following is the C type for a linked list node:

```
struct list_node {
        int data;
        struct list_node * next;
};
```

Complete each of the following sub-questions, using this node type.

(a) Write a C function that takes a pointer to the first node in a linked list and frees the memory used by the linked list. The prototype of your function will be:

```
void free_list(struct list_node * head);
```

[10 marks]

(b) Write a C function that takes a pointer to the first node in a linked list and returns a pointer to a reversed version of the list. That is, the first element becomes the last, the second element becomes second last and so forth. You should return the nodes of the original list in reverse order rather than creating a new list or simply changing the *data* fields stored in the list. Briefly describe your approach. The prototype for your function will be:

```
struct list_node * reverse_list(struct list_node * head);
```

[15 marks]

(c) Finally, a common programming error when writing code that works on linked lists is to accidentally introduce a cycle into the list. A cycle exists when a node in the list points either to itself or to an earlier node in the list. Code that traverses a list that contains a cycle often gets stuck in an infinite loop, because it never reaches the end of the list. Write a C function that returns 1 if the list data structure contains a cycle, and 0 otherwise. Briefly describe your approach. The prototype of you function will be:

```
int contains_cycle(struct list_node * head);
```

Your function should finish in finite time. That is it should not get stuck in an infinite loop if the list contains a cycle.

[25 marks]

2. You are building software to track the free parking spaces in a multi-storey car park. Each level of the car park has *N* spaces, and each space has a number between 1 and *N* inclusive. In addition the car park has *L* levels numbered 1 to *L* inclusive. Thus to uniquely identify a given parking space, one must specify both the level number and the space number within the level.

   To track the free spaces you need to build a set abstract data type (ADT) which is capable of recording which spaces are occupied and which are free. The number of levels (*L*) and number of spaces within each level (*N*) are fixed and will not change during the execution of your program.

   Write a C abstract data type that represents the set of free parking spaces in a given multi-story car park. For performance purposes, you must be able to add (or remove) a parking space to (or from) the set in worst case constant time. You should also be able to find whether or not a given parking space is occupied in constant time, given the space's level number and space number. Finally, you should provide a function for freeing any allocated memory when the set of parking spaces is no longer needed.

   Your software will have to run on a low powered embedded system, so processing power and memory are limited. Therefore you should take a reasonable amount of care that your ADT does not use much more memory than necessary. You are required only to build the set ADT, not any other software dealing with the tracking of parking spaces.

   [50 marks]

3. Although the most common representation of character strings is as an array of characters, 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 [5 marks]

   b) Take a normal null-terminated array C string as a parameter, and returns a new linked list-based string of type *mystring.* [5 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. [10 marks]

   d) A search/replace function that takes three *mystrings* as parameters: a piece of *text*, a *search* string and a *replace* string. The function should search the *text* string for the *search* string, and wherever it finds an instance of the *search* string within the text, it should replace the *search* string with the *replace* string. Finally the function should return the modified *text* string. [30 marks]

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