IMPERIAL COLLEGE LONDON

DEPARTMENT OF ELECTRICAL AND ELECTRONIC ENGINEERING
EXAMINATIONS 2008

ISE PART I/EEE PART II

Corrected Copy

SOFTWARE ENGINEERING: INTRODUCTION, ALGORITHMS AND DATA STRUCTURES

Friday, 13 June 10:00 am

Time allowed: 1:30 hours

There are THREE questions on this paper.

Answer Question ONE and ONE other question.

This exam is OPEN BOOK.

Question One carries 40% of the marks; Questions Two and Three each carry 60%.

Any special instructions for invigilators and information for candidates are on page 1.

Examiners responsible   First Marker(s) :   C. Bouganis

                            Second Marker(s) :   L.G. Madden

This page is intentionally left blank.

Special information for invigilators:

Students may bring any written or printed aids into the examination.

Information for candidates:

Marks may be deducted for answers that use unnecessarily complicated algorithms.

The Questions

1. **[Compulsory]**

    a) Figure 1.1 shows a C++ function that calculates the value of the function described in equation (1.1), for a value of n where n is integer.

$$f(n) = \begin{cases} f(n-1)+n-1 & \text{if } n > 1 \\ 0 & \text{if } n = 1 \end{cases} \qquad (1.1)$$

Identify five errors in the C++ code shown in Figure 1.1.

```
float calculateF (n) {
    int result=0;
    for (i=2; i < n; i=i+2)
        result = result + (i-1);
    return result;
}
```

Figure 1.1 calculateF function.

[3]

    b) Write a C++ recursive function that performs the calculation described in part (a).

[3]

    c) i)   A set of numbers is inserted in an ordered binary tree (ascending ordered tree). Draw a tree for the following set assuming that the elements in the set are inserted in the order shown.

$\{20, 30, 10, 25, 40, 35, 45\}$

[2]

        ii) An alternative data structure to a tree is a list. Draw an ordered list for the set of part (i) assuming that the elements in the set are inserted in the order shown.

[1]

        iii) Draw the final data structures of parts (i) and (ii) after a deletion operation of value 30 has been performed.

[1]

        iv) Comment on the average number of operations that is required to delete an item from an ordered binary tree and from an ordered list. Assume that the information is stored to the nodes of the tree. (Hint: Consider the search phase and the actual delete phase separately).

[2]

d) Construct a parse tree for the following expressions, assuming the normal priorities of the operators:

   i) $3+5/2$ [1]

   ii) $2+5*6/7+3$ [1]

e) Consider the C++ code segment in Figure 1.2. With justification, state the values of variables x and y after the code segment is executed.

```
int x=1;
int y=2;
int *px;
int *py;
px = new int;
py = &y;
*py = 10;
*px = x;
*px = *px + 1;
py = px;
*py = *py + 1;
```

Figure 1.2 Code segment.

[3]

f) Figure 1.3 shows the type declaration for a dynamic linked list of integers in C++.

```
class Node {
    public:
        int data;
        Node * next;
};

typedef Node * NodePtr;
NodePtr hdList = NULL;
```

Figure 1.3 Linked list declaration.

   i)   Write a C++ recursive function that takes as input the *hdList* pointer and returns the number of items in the list that their data field has zero value, *i.e. data*=0.

[2]

   ii)   Write a C++ function that takes as input the *hdList* pointer and using an iteration performs the same operation as in part (i).

[1]

2. RNA is a sequence of nucleotides. There are four types of nucleotides: T, U, C, and G. Figure 2.1 illustrates an example of an RNA sequence.
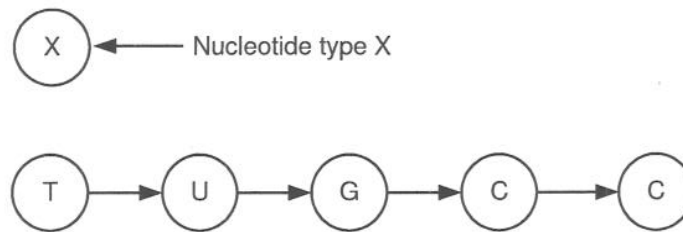

Figure 2.1 RNA sequence "TUGCC".

a) Define a structure *RNAnode* capable of representing a nucleotide in an RNA sequence.

[5]

b) Write a function that takes as inputs a pointer to the start of an RNA sequence and a nucleotide type b, and returns the number of appearances of nucleotide of type b in the sequence.

[5]

c) Write a function that takes as input a pointer to the start of an RNA sequence and checks if the following nucleotide sequence "TU" appears in the RNA sequence.

[5]

d) Write a function that takes as input a pointer to the start of an RNA sequence and returns the number of appearances of the following nucleotide sequence "TU" in the RNA sequence.

[5]

3.   It is assumed that an organism evolves only through mutations. Figure 3.1 illustrates the sequence of evolution of such an organism. Each node represents an instance of the organism during its evolution. Every arrow in the figure corresponds to a mutation. It is assumed that the organism can evolve to one instance or to two instances at any given time. Every node has a unique number *id*.
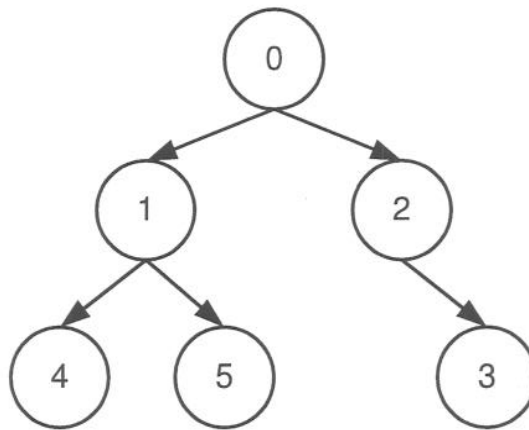

Figure 3.1 Evolution tree

a)   Define a structure *Node* capable of representing a node of the structure shown in Figure 3.1.

[5]

b)   Write a recursive function/procedure that takes as inputs the pointer to root node ($id = 0$) and an input variable *id*, and returns the number of instances of the organism that have been evolved from node *id*. For example, for the tree in Figure 3.1, for *id=2* the function/procedure should return value 1, where for *id=0* the function/procedure should return value 5. You may use extra arguments.

[5]

c)   Write a recursive function/procedure that takes as inputs the pointer to root node ($id = 0$) and an integer N, and returns the number of instances of the organism that have been evolved from node 0 with N mutations. For example, for the tree of Figure 3.1, if $N = 2$, then the function should return the value 3. You may use extra arguments.

[5]

d)   Write a recursive function/procedure that returns the number of instances that have been evolved to one only instance, *i.e.* nodes that have only one child. The function should take as input the pointer to root node ($id = 0$). You may use extra arguments.

[5]