NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY
FACULTY OF APPLIED SCIENCE
COMPUTER SCIENCE DEPARTMENT
JUNE EXAMINATIONS 2004

SUBJECT: DATA STRUCTURES AND ALGORITHMS
CODE: SCS2103

INSTRUCTION TO CANDIDATES

Answer any five questions.
Each question carries 20 marks
Total marks 100
Write all code in the C programming language

Time: 3 hours

QUESTION ONE

Implement the following operations for a static list data structure.

(i) Front, which returns the contents of the first element in the list; [5]
(ii) DeleteElem, which deletes a specific element from the list, [5]
(iii) AddToFront, which connects an element to the front of the list; [5]
(iv) Append, which adds an element to the end of the list. [5]

QUESTION TWO

Write a function, which returns the concatenation of two lists L1 and L2, that is, a list containing copies of all the nodes of L1 followed by copies of all the nodes of L2. Note that this function does not destroy either L1 or L2. All lists must be represented dynamically.

[20]

QUESTION THREE

Using a variable to distinguish between a circular queue being empty or full, write

(i) Insert, [4]
(ii) Delete, [4]
(iii) IsEmpty, [4]
(iv) IsFull and [4]

(v) Display functions. [4]

QUESTION FOUR

Let $L = (a_0, a_1, \ldots, a_{n-1})$ be a linear list represented in the array, `int v[n]` using the mapping: the i'th element of L is stored in `v[i]`.

(i) Write an algorithm to make an in-place reversal of the order of elements in `v`. That is, the algorithm should transform `v` such that `v[i]` contains the `(n - 1 - i)`'th element of L. The only additional space available to your algorithm is that for simple variables. The input to the algorithm is `v` and `n`.
[10]

(ii) How much time does your algorithm take to accomplish the reversal?
[5]

(iii) Use the Big O notation to specify this time. [5]

QUESTION FIVE

Implement the following operations for a dynamic binary search tree data structure.

(i) Insert, to insert an element into the tree; [4]
(ii) Count to return the number of elements in the tree; [4]
(iii) Preorder, to perform a preorder traversal on the tree; [4]
(iv) Inorder, to perform an inorder traversal on the tree; [4]
(v) Postorder to the perform a postorder traversal on the tree. [4]

A treenode, and a treeptr should be defined as follows:

```
typedef struct tn {
   int data;
   struct tn* left;
   struct tn* right;
} treenode, *treeptr;
```

QUESTION SIX

Implement the following operations for both a static stack representation, and a dynamic stack representation:

(i) Push, to add an element to the top of the stack;
[5]

(ii) Pop to return the topmost element in the stack and also to remove it from the top of the stack, [5]

(iii) Top, to return the topmost element in the stack without removing it from the stack. [2]

(iv) Compare the performance of the two representations. [8]

QUESTION SEVEN

Provide a detailed explanation of the Big O notation. Include in your explanation, the need for a model of computation.

[20]

END OF QUESTION PAPER

GOOD LUCK!