EASTERN UNIVERSITY, SRI LANKA

SECOND EXAMINATION IN SCIENCE 2001/2002

FIRST SEMESTER (April, 2002)

CS 201 - Data Structures & Design of Algorithms

Answer All Questions

Time: 2 Hours

Q1

Describe briefly the ***Array*** data structure.

A lower triangular matrix is a square matrix $A=(a_{ij})$ in which $a_{ij}=0$ for $i<j$. It is written as

$$A=\begin{pmatrix} a_{11} & & & & & & \\ a_{21} & a_{22} & & & & & \\ a_{31} & a_{32} & a_{33} & & 0 & & \\ \vdots & \vdots & \vdots & & & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & \cdots & a_{nn-1} & a_{nn} \end{pmatrix}$$

Define a sequential allocation for such matrices.

Write down the number of elements in the lower triangular part of an M*M lower-triangular matrix.

Write a C++ program to read an M*M lower triangular matrix into a one-dimensional array as a row-major representation, and find the row-sums.

An upper-triangular matrix is a square matrix $A=(a_{ij})$ in which $a_{ij}=0$ for $i>j$. Show how you would represent into a one-dimensional array as a column-major representation. (Hint: Consider the transpose)

Q2

Define and implement ***ADT Stack*** data structures using linked list to represent the list of elements.

(a) Write a C++ program that reads an integer number and to output whether the input number is a palindrome or not.

(b) Define ***ADT Queue structure*** and implement it using two stacks.

Q3

Describe briefly the ***Backtracking*** technique with a suitable example. Suppose that **S** be a given set of integers and **M** be a given integer number. You are required to find all-possible subsets of **S** in which the sum of elements of a subset must be equal to **M**.

Write an algorithm to solve the above problem using backtracking technique.

Trace this algorithm for the following set of data:
s={1, 7, 8, 14} and M=15.

Show how you would modify your algorithm to find the sum to be less than or equal to M.

Q4

(a) Describe the ***Bubblesort*** algorithm to sort any given list of numbers.

Give a complexity analysis of the ***Bubblesort*** algorithm.

(b) Describe briefly ***Divide-and-Conquer*** technique with a suitable example.

Describe the ***Quicksort*** algorithm to sort given list of numbers.

Give a complexity analysis of the ***Quicksort*** algorithm.

Trace the above sorting algorithms for each of the following lists of numbers:

(a) 1, 1, 1, 1, 1

(b) 1, 2, 3, 4, 5

(c) 9, 8, 7, 6, 5

(d) 5, 8, 3, 7, 1