EASTERN UNIVERSITY, SRI LANKA

SECOND EXAMINATIN IN SCIENCE 2003/2004 (Repeat)

FIRST SEMESTER (November/December, 2004)

CS201 Data Structures and Design of Algorithms

Answer all questions

Time allowed: 2 Hours

Q1

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

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

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n-1} & a_{1n} \\ & a_{22} & a_{23} & \cdots & a_{2n-1} & a_{2n} \\ & & a_{33} & \cdots & a_{3n-1} & a_{3n} \\ & & & & \vdots & \vdots \\ & & \mathbf{0} & & \vdots & \vdots \\ & & & & a_{n-1n-1} & a_{n-1n} \\ & & & & & a_{nn} \end{pmatrix}$$

Define a sequential allocation for such matrices.

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

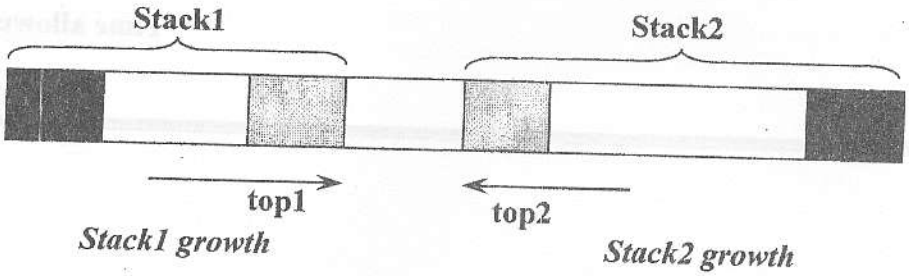
Write a C++ program to read an **M x M** upper triangular matrix into a one-dimensional array as a row-major representation, and find the row-sums.

An lower-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 the ADT stack.

An array can be used to store two stacks, one growing up from the left end, and the ot
growing up from the right end.



Write down the conditions for stack1 and stack2 to become empty.

Write down the conditions for stack1 and stack2 to become full.

Implement the class DualStack whose declaration is given by

```
Const int MaxDualStackSize=200;
Class DualStack
{
   private:
        int top1, top2;
        DataType StackStorage[MaxDualStackSize ];
   public:
            DualStack(void);
            void Push(DataType elt, int n);  // push elt on stack n
            DataType  Pop(int n);                    // pop from stack n
            DataType Peek(int n);                    // peek at stack n
            int StackIsEmpty(int n);          // is stack n empty?
            int StackIsFull(int n);           // is stack n full?
            void  ClearStack(int n);
};
```

Write a main program that can do the following:

(i) read a sequence of integers and to push the even ones on stack1 and the odds on stack2.

(ii) print an appropriate message if any stack becomes full.

(iii) print the contents of each stack with an appropriate message if a stack is empty.

Q3

Describe briefly the ***Backtracking*** technique with a suitable example. Suppose that **S** is a given set of integers and **M** is 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 = {4, 8, 12, 16} and M=20.

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 **Merge*sort*** algorithm to sort given list of numbers.

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

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

(a) 2, 2, 2, 2, 2, 2

(b) 2, 5, 6, 8, 9, 12, 15

(c) 9, 7, 4, 3, 2, 1

(d) 5, 7, 1, 2, 9