**Mid Term Examination**

Data Structures

Summer 2011

**Time: 2 hr 30 min**

**Q#1** (10 marks) Consider the following function:

int solve(int a, int b)

{

if(a==0)

return b;

if(a>b)

return solve(a-b, b);

else

return solve(b-a, a);

}

Show the output and the tracing of the activation stack for ***recursiveFunction(255,867)***.What is the objective of this function?

**Q#2.** (10 marks) What is the time complexity of the following pieces of code in terms of big-Oh? Show all working.

|  |  |
| --- | --- |
| int i=0;  while(i<=n){  for(int j=n; j>=i ; j--){  for(int k=1; k<100; k=k\*2)  cout<<i+j+k;}  i++;  } | int i=n;  while(i>0){  for(int j=n; j>=i ; j=j-1)  cout<<i+j;  i=i/2;  } |

**Q#3.** (10 marks) Consider the following scenario. We are asked to design a system for an egg supply firm. They use the following resources to keep their eggs. Firstly, eggs are placed in trays that have 10 rows and 10 columns each. Then these trays are stacked on top of each other. The maximum height of a stack is 30 trays. Afterwards, stacks are packed in boxes such that each box contains 6 stacks. These boxes are then delivered on trucks. Each truck can store 100 boxes. Furthermore, each truck belongs to a warehouse. Each warehouse has 15 trucks and there are 20 warehouses in total.

Each egg is given an integer code. We have to keep track of all the eggs. Describe which data structure we should use to be able to retrieve a particular egg as efficiently as possible. Also write C++ code for:

a) Defining your particular data structure.

b) Accessing the egg in the 6th row and 7th column of the 4th tray in the 3rd stack, in 34th box, in 12th truck of the 19th warehouse.

**Q#4** (5+15 marks) a) Write C++ code for designing an array based stack. You can take the array to be of fixed size, e.g., 100. Also give the implementation of all its relevant member functions.

b) A stueue is a data structure that is capable of performing the functionalities of both the stack and the queue simultaneously. It has the following three functions:

Push: Add a new element to stueue.

Pop: Return the element that was added the last of all elements currently present in the stueue and delete it from the stueue.

Deque: Return the element that was added before (first) all the other elements currently present in the stueue and delete it from the stueue.

Using only the stack class implemented in part (a) of this question, implement Stueue. You must not use any other data structure than the stack you just implemented.

Hint: You may use multiple instances of your stack class.

**Q#5** (5+15 marks) a) Write C++ code for designing a doubly link list. Create all the relevant classes with all the relevant data members and member functions. Just give the prototypes of the member functions and not their implementations.

b) Write a function that takes two sorted singly link lists and returns a new sorted singly link list (head pointer) which has all the elements of both the original lists in a sorted order. You may assume that this function is a friend of the original link list class.