THE OPEN UNIVERSITY OF SRI LANKA
B.Sc DEGREE PROGRAMME: LEVEL 03
CLOSED BOOK TEST - 1: 2010
CPU 1142: DATA STRUCTURES AND ALGORITHMS

DURATION: **ONE AND HALF HOURS** (1 ½ HOURS)

Date: 05th March, 2010 Time: 4.00 pm – 5.30 pm

Answer ALL questions.

Q1.

a)

i. What is a *Data Structure*?

ii. List down the three (03) basic operations that can be performed with Data Structures.

b) What is a *List*?

c) Distinguish between *Array Implementation of List* and *Pointer Implementation of List*.

d) Use the following C language declaration of *Array Implementation of List* to answer the questions from d (i) to d (iii).

```
1     #define nodes 100
2     struct node
3     {
4           int info;
5           int next;
6     }
7      struct node strArray[nodes];
```

i. Graphically explain the functionality of the above C code.

ii. Write C program code for the following functions:

`int getNode()` – Remove a node from the available list and return a pointer to it.

`void insertAfter(int p, int x)` – Insert variable x into a node following the `node(p)`.

`void freeNode(int p)` – Release a node to the available list.

iii. Write an algorithm to delete the node following the `node(p)` and store its contents in variable character x.

Q2.

a)

i. What is a *Linked List*?

ii. What are the features of a Linked List? Briefly explain them?

b) What is a *Stack*? Why it is called as a LIFO data structure?

c) A Stack has the following basic operations.

`POP(S)` – remove the top element of the Stack S
`PUSH(x, S)` – insert element x at the top of the Stack

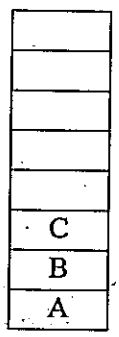


Figure 1: Current state of the stack

i. *Figure 1* shows the current state of the Stack. Graphically show the following operations that can be performed on the above Stack. Indicate the `top` pointer in each state of the Stack.

```
PUSH(F, S);
PUSH(I, S);
POP(S);
PUSH(D, S);
POP(S);
POP(S);
```

ii. Write the C language code to perform the `PUSH` and `POP` operations.

Q3.

a) Use the following Tree (Figure 2) to answer the questions from a (i) to a (vi).

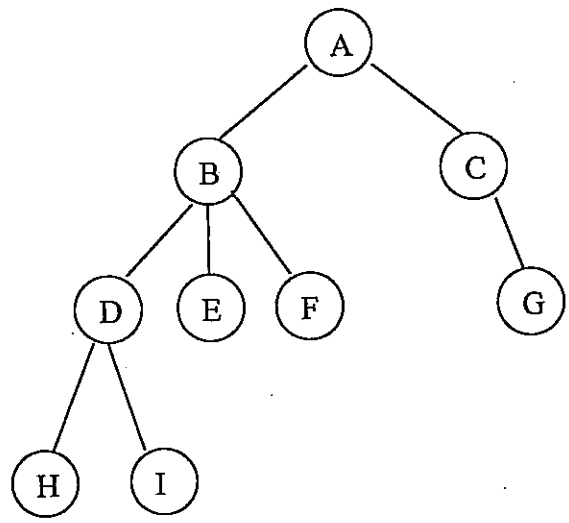


Figure 2: Structure of the Tree

i. How many *levels* are there in the above Tree?

ii. What is the *level* of node F?

iii. What are the *terminal nodes* in the above tree?

iv. Distinguish between the *degree of a node* and *degree of a tree.*

v. What is the *degree* of the above Tree?

vi. What are the *degrees* of node B and node C?

b) What is a *Binary Tree*? Explain by using two (02) suitable examples.

c)

i. What is a *Strictly Binary Tree*?

ii. Give two (02) examples for Strictly Binary Tree. They should have more than two (02) levels.

THE OPEN UNIVERSITY OF SRI LANKA
B.Sc DEGREE PROGRAMME: LEVEL 03
CLOSED BOOK TEST - 02: 2010
CPU 1142: DATA STRUCTURES AND ALGORITHMS

DURATION: **ONE AND HALF HOURS** (1 ½ HOURS)

Date: 30th April, 2010 Time: 4.00 pm – 5.30 pm

Answer ALL questions.

Q1.

a) What are the two types of *recursion*? Briefly explain them.

b) Write a recursion function to print the square values of numbers from a given positive number upto 1. The output should be as follows, if the given number is 5.

```
square value of 5 = 25
square value of 4 = 16
square value of 3 = 9
square value of 2 = 4
square value of 1 = 1
```

c) What are the two types of *sorting methods*? Briefly explain them.

d) How many passes are required to sort *n* number of integers by using the *bubble sort*?

e) Explain how the *bubble sort* algorithm works on the following set of integers, when sorting them in ascending order.

55, 82, 24, 32, 64, 12, 8, 20

f) Compare and contrast the *bubble sort* algorithm with the *quick sort* algorithm.

Q2.

a) What is a *binary search tree*? Briefly explain.

b) Construct a binary search tree for the following set of integers. Answer the questions from (c) to (e) by using the binary tree that you have constructed.

13, 15, 7, 9, 2, 13, 20, 12, 7, 2, 16, 15, 4, 11, 20

c) What will be the output, when you traverse the tree in the following orders?

i. Preorder (NLR)
ii. Inorder (LNR)
iii. Postorder (LRN)

d) What is the *depth of the tree*?

e) What are the *leaf nodes* and *non leaf nodes* of the tree?

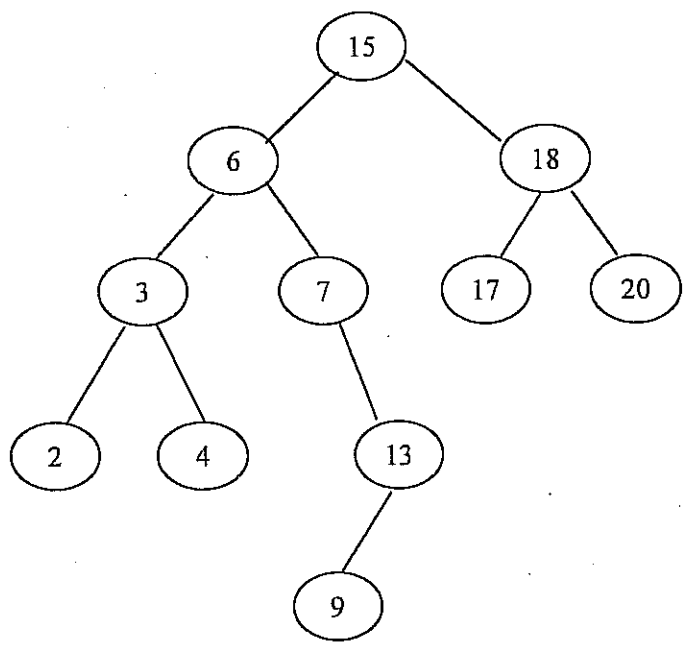
Q3.

a) What is meant by the *Binary Search*?

b) State how binary search can be applied for the following array, to search the element with the value '12'? Clearly state the steps that are involved.

2	8	11	12	13	18	20	22	25

c) What will be the result if the node with value 6 is removed from the following binary tree? Clearly draw the resulting diagram of the below tree structure.



d) What is a *heap*?

e) What are the two types of *heap*? Briefly explain them.