THE OPEN UNIVERSITY OF SRI LANKA
DEPARTMENT OF MATHEMATICS & COMPUTER SCIENCE
B.Sc. DEGREE PROGRAMME : LEVEL 03
CPU1142- DATA STRUCTURES & ALGORITHMS
FINAL EXAMINATION – 2015/2016

DURATION:Two Hours (2 Hours)

Date: 13.01.2017 **Time: 1.30p.m. – 3.30 p.m.**

Answer FOUR (4) Questions ONLY.

QUESTION 01

1) Write the definition of a **Minimum Spanning Tree**?

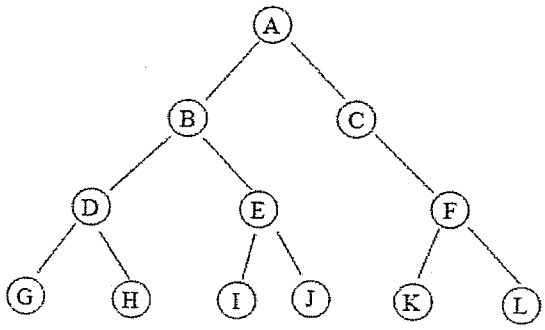
2) Write codes in C programming language to do the following.
   a. Create a data structure to implement a **queue**. The structure should contain variables to store the **front** and **rear** positions of the **queue** and an **array** to hold numeric data of elements of type **int** in the **queue**.
   b. Using the structure that you created in section (a) above, declare a queue with the name "**myqueue**".
   c. Write a function in C to **delete** a data item from "**myqueue**". Use **int deleteQueue()** as the function header (check queue underflow condition before you delete a data item).
   d. Write a function in C to **insert** a data item into "**myqueue**". Use **void insert (int x)** as the function header. (Check the queue overflow condition before you insert a data item).

3) Using **big O** notation, determine the running time of the following C program section. State any assumptions you make.

```
for(i=1; i<n; i++)
for(j=i; j<n; j++) printf("(%d,%d)\n",i,j);
```

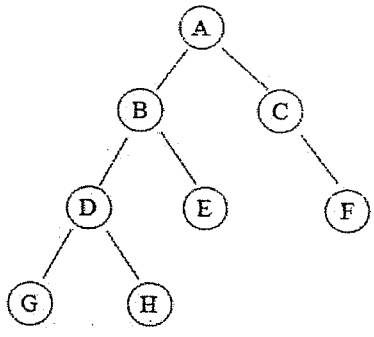
4) What is the task of the **malloc()** function?

5) Define a "**complete binary tree**". Is the following tree a complete binary tree? Give reasons.



QUESTION 02

1) Define a "**Strictly Binary Tree**". Is the following tree a strictly binary tree? Give reasons.



2)

a) Construct a binary search tree for the following set of integers.
40, 10, 15, 60, 05, 20, 50, 70, 90, 02, 65, 08

b) What will be the output when you traverse the above binary tree, which you constructed in part (a) above in the following orders?

i. Pre-order
ii. In-order
iii. Post-order

c) What is the depth of the binary search tree which you constructed in part (a) above?

d) What are the **leaf** nodes of the binary search tree which you constructed in part (a) above?

3) Describe the following terms with respect to **GRAPH** data structure.
   a) Adjacent Vertices
   b) Cycle
   c) Connected Graph
   d) Path

4) State the two main types of sorting algorithms.

QUESTION 03

1) Write a function in C programming language to implement the **Insertion Sort**. Use **void insertionSort (int numbers[], int array_size)** where the array named "**numbers[]**" has the elements to be sorted and "**array_size**" gives the number of elements.

2) What is the running time of the insertion sort (use **big O** notation)?

3) What will be the output of the following function if we pass "**6**" as the value for **n**?

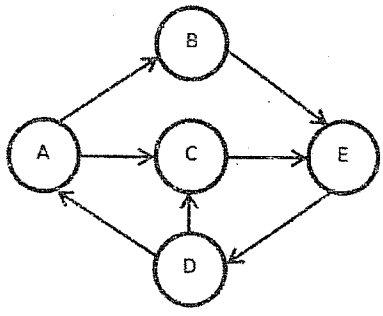
```
int fact(int n)
{
if (n == 0)
return 1 ;
else
return n * fact(n-1) ;
}
```

4) Write codes in C programming language to do the following.
   a) Using **array implementation** of a stack, create a data structure of a stack (array size is **100**).
   b) **Push** an element to the **stack**.

QUESTION 04

1) What is a Weighted Graph?

2) Show the **multi-list representation** of the following graph.



3) Using the graph of the part (2) above, show the order of vertices visited in the "**Depth first**" and "**Breath first**" traversals. Select "**A**" as the starting node.

4) Graphically show the steps of sorting the following data set by using **selection sort**.

50	30	60	20	40	70	10

QUESTION 05

1) What is **Divide and Conquer Method** and give one example for that?

2) Write codes in C programming language to do the following.
   a) Using **pointer implementation**, create a data structure of a circular linked list.
   b) **Delete** an element at the **beginning** of the circular linked list.

3) Consider the following expression in **Infix** form and convert it into the **Postfix** form. Clearly show the 7 **steps** required for the conversion.

   (A-B)*C/D+E*F/G$H

4) Following is a function in C programming language for some operation.

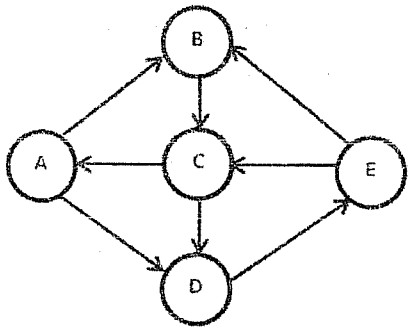
```
void insert (int x)
{
if(q.front == (q.rear+1)% MAXSIZE)
{
printf("queue overflow \n");
exit(1);
}
q.rear = (q.rear+1)% MAXSIZE;
q.items[q.rear]=x;
}
```

By analyzing the above function in C programming language, answer the following questions.

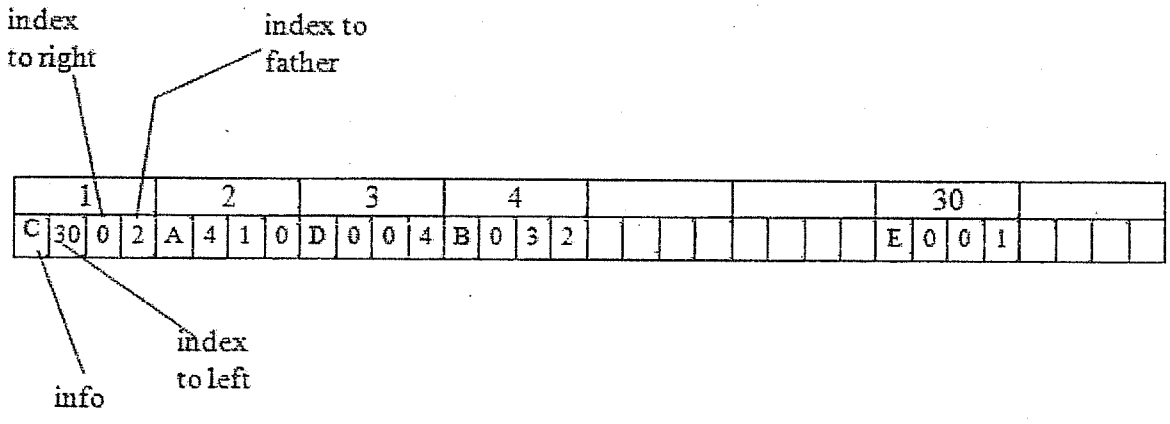
   a) What is the data structure that this function belongs to?
   b) What type of implementation method is used to create the data structure that you mentioned in section (a)?
   c) Which type of operation can perform using the above function?

QUESTION 06

1) What is "**Greedy**" method and give one example for that?
2) Show the **adjacency-list** representation of the following graph.



3) Array representation of a binary tree is as follows.



a) Draw the relevant binary tree according to the above array representation.
b) Mention the degree of each node of the binary tree which you constructed in part (a) above?

4) Consider the running times **O(f(n))** and **O(g(n))** where

$$f(n) = n^2 \text{ if } n \text{ is even}$$
$$n^6 \text{ if } n \text{ is odd}$$

$$g(n) = n^4 \text{ if } n \text{ is even}$$
$$n^3 \text{ if } n \text{ is odd}$$

Consider that **f(n)** and **g(n)** are placed sequentially.

a) Calculate the time complexity for both **odd n** and **even n** separately.
b) What is the name of the **rule** that required for calculate the time complexity.