THE OPEN UNIVERSITY OF SRI LANKA
B.Sc. DEGREE PROGRAMME: LEVEL 03
CLOSED BOOK TEST - 2: 2010/2011
CPU1142: DATA STRUCTURES AND ALGORITHMS

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

Date: **29th April, 2011** Time: **4.00 pm – 5.30 pm**

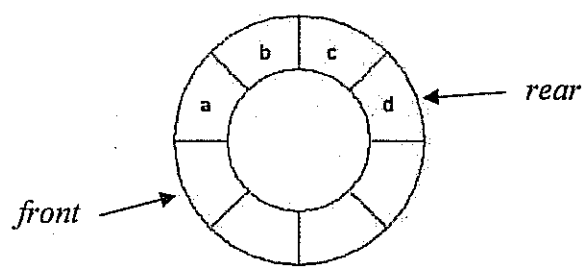
Answer ALL questions.

Q1.

a) There are two basic operations associated with a Circular Queue known as `enqueue` and `dequeue`.

`enqueue(X)` – Adding a new item X in to the Circular Queue
`dequeue()` – Removes the item from the Circular Queue

The following figure shows the current state of the Circular Queue.



Graphically show the following operations that can be performed on the above Circular Queue. Indicate the `rear` and `front` pointers in each state of the Circular Queue.

```
enqueue(e);
enqueue(f);
enqueue(g);
dequeue();
enqueue(h);
enqueue(i);
```

b) Consider the following **palindrome** which is a string that reads the same forward as backwards.

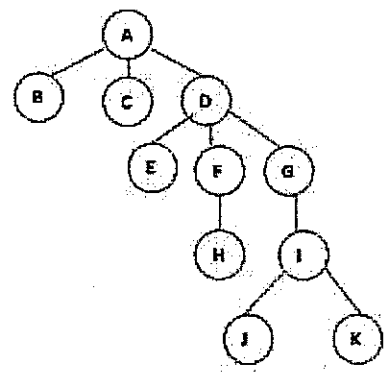
Air an aria

Using only a fixed number of stacks and queues, write an algorithm to determine whether the string is a palindrome.

Assume that, the string is read one character at a time. The algorithm should output TRUE or FALSE as appropriate.

Q2.

a) Consider the following Tree to answer the questions from a) i. to a) vi.



i. Mark the levels of the Tree.

ii. What are the Sub Trees of the node D?

iii. What are the descendants of node G?

iv. What is the depth of the Tree?

v. What is the degree of the node D and node G?

vi. What is the degree of the Tree?

b) Construct a Binary Search Tree for the following set of integers.

25, 26, 12, 32, 6, 49, 8, 16, 33, 2, 11, 21, 9, 15, 53, 64

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

i. Preorder (NLR)

ii. Inorder (LNR)

iii. Postorder (LRN)

Q3.

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

2, 1, 6, 4, 2, 3, 5

b) Consider the following Bubble sort algorithm.

```
void bubbleSort(int a[], int n)
{
      for (i=0; i<n-1; i++) {
         for (j=0; j<n-1-i; j++)
             if (a[j+1] < a[j]) {
```

```
                tmp = a[j];
                a[j] = a[j+1];
                a[j+1] = tmp;
            }
        }
}
```

The above implementation has the following inner for loop:

```
for (j=0; j<n-1-i; j++)
```

Consider the effect of replacing this with the following statement:

```
for (j = n-1; j>i; j--)
```

Would the new implementation work correctly? If the new implementation is working correctly rewrite the above algorithm.

c) Compare the *Bubble sort* algorithm with the *Quick sort* algorithm.