



United International University (UIU)

Dept. of Computer Science & Engineering (CSE)

Final Exam Year: 2018

Trimester: Fall

Course: CSI 217 Data Structures, Marks: 40, Time: 2 hours

There are **FOUR** sets of questions. Answer **ALL** of them.

1. a. Write a function that prints the values of the first and last node of a doubly linked list. It also counts total number of nodes in the doubly linked list.

Assume that, the node is a structure that looks like as follows. Also assume that, there is data already inserted in the linked list, and 'head' is the pointer to the start of the linked list and 'last' is the pointer to the end of the linked list. You just have to write a function called **Print_And_Count_Node()** to do the task. [2.5]

```
struct node{
    struct node *prev;
    int value;
    struct node *next;
};
struct node* head;//points to the start of the doubly linked list
struct node* last;//points to the end of the doubly linked list

void Print_And_Count_Node(){

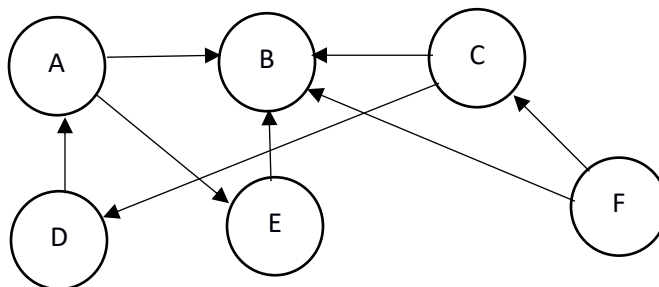
}
```

- b. Write a code segment to **insert** an element in the **i) front and ii) end** of a linear circular linked list. [2. 5]

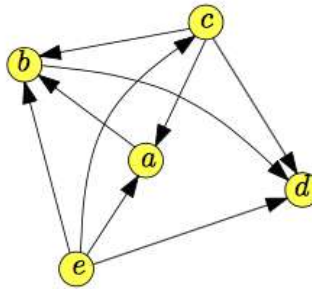
- c. Design a queue using linked list. Consider function name **insert()** and **delete()**. [2.5]

- d. Implement the **push()** and **pop()** operations of a stack (stack with array) [2.5]

2. Consider the following graph.



- a. Draw the adjacency matrix of the above Graph. [2]
- b. Simulate the above graph using BFS. Show each and every steps in details. [4]
- c. Show the mechanism of topological ordering algorithm for the following directed acyclic graph. [4]



3. a. Construct a **binary search tree** inserting the following numbers sequentially:
10, 55, 100, 20, 23, 15, 65, 30, 28 [4]
- b. Write the preorder, postorder and inorder traversal sequence of your constructed tree in Question 3(a). [3]
- c. Delete node 55, then 65, then 15 (**sequentially**) from your constructed tree in Question 3(a). [3]
4. a. Convert the infix expression: $(a+b-c*d)/a*(e-f/b*e)-f*g$ into corresponding postfix expression using stack. Then evaluate the postfix expression for the given value: $a = 5, b = 7, c = 2, d = 1, e = 5, f = 20, g = 2$ and compare the result. [4]
- b. Insert the following items in a Hashtable (**size 10**) using **Linear probing** as collision avoiding technique. [3]
2, 4, 18, 14, 11, 45, 50, 55, 12
Hash function **H** is defined as below:

```

int H (int x){
    return x mod 10 ;
}

```
- c. Construct a binary tree from the given traverse result: [3]
Postorder : g h d l e b j f c a
Inorder : g d h b e l a f j c