DSA_Quiz-3_9/3/2021 (CE245-DSA-4CE1)

Includes singly linked list, circular linked list

* This form will record your name, please fill your name.

1. Given code deletes the node which doesn't follow the ascending order sequence from this linked list : 3>>>4>>>5>>>6>>>7>>>9>>>8>>>10...

   Select correct statements for a1,a2,a3,and a4.

   node structure is defined using this code:

```
struct node
{
    int data;
    struct node * link;
} *head, *p;
```

   Code to remove node is::

```
  struct node *C= head;
     while(C->data <= a1)
        {
             p=a2;
            C=a3;
       }  a4;
```

   (4 Points)

   - a1=C->data->link, a2=C, a3= C->link, a4=p->link=C->link
   - a1=C->link->data, a2=C, a3= C->link, a4=p->link=C->link
   - a1=C->link>data, a2=C->link, a3= C, a4=p->link=C->link
   - a1=C->link>data, a2=C, a3= C->link, a4=C->link=P->link

2. Consider the function shift_last_node_to_first() to shift the last node of singly linked list to first node.

```
struct node
{
    int data;
    struct node * link;
} *head;

void shift_last_node_to_first()
{
      struct node * t = head, *p;
     while(a1)
            {
                  a2;
                    t = t->link;
            }
           a3 = head;
           a4= t;
           a5;
}
```

Write the statements in place of a1,a2,a3,a4 and a5.

Answer format example : a1=t!=NULL,a2=t->link->link
(Do not give any spaces and separate a1,a2 with comma as format given. If format of answer is not followed, marks will not be awarded.)
(4 Points)

3. State True or False: In linked list, it is not necessary that all elements are arranged in physical contiguous memory.
(1 Point)

- ◯ True
- ◯ False

4. Linked list is :
(1 Point)

- ◯ Linear Data Structure
- ◯ Non Linear Data Structure

5. What is the output of following function for start pointing to first node of following linked list?
1->2->3->4->5->6

```
void fun(struct node* start)
{
  if(start == NULL)
    return;
  printf("%d  ", start->data);

  if(start->next != NULL )
       fun(start->next->next);
  printf("%d  ", start->data);
}
```

(4 Points)

- ◯ 1 4 6 6 4 1
- ◯ 1 3 5 1 3 5
- ◯ 1 3 5 5 3 1
- ◯ 1 2 3 5

6. In Round-robin scheduling of processes in OS, each ready task runs turn by turn only for a limited time slice. After running last task, again from first task all tasks run turn by turn. Which data structures are appropriate to implement this scheduling?
(1 Point)

- [ ] Stack
- [ ] Circular Queue
- [ ] Singly Linked List
- [ ] Circular Linked List
- [ ] Simple Queue