Summary in Graph

Exam Summary (GO Classes Test Series 2024 | Data Structures | Test 1)

Qs. Attempted:	**12** (5 + 7)	**Correct Marks:**	**19** (5 + 14)
Correct Attempts:	**12** (5 + 7)	**Penalty Marks:**	**0** (0 + 0)
Incorrect Attempts:	**0** (0 + 0)	**Resultant Marks:**	**19** (5 + 14)

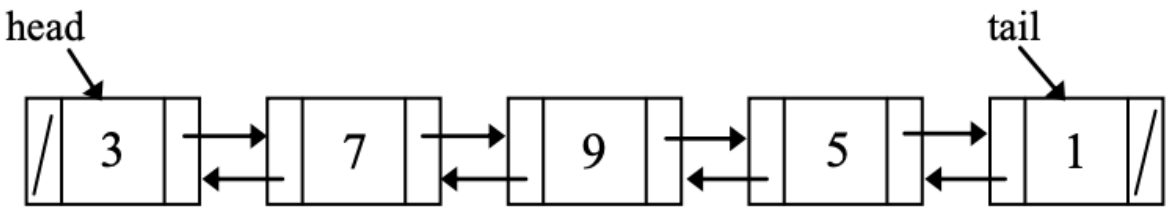
Total Questions:	**15** (5 + 10)
Total Marks:	**25** (5 + 20)
Exam Duration:	**45 Minutes**
Time Taken:	**45 Minutes**

EXAM RESPONSE | EXAM STATS | FEEDBACK

Technical

Q #1 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | DS

Given a doubly linked list where each node has two references (prev and next): one that points to a previous node and another that points to a next node. Assume the linked list below and provide the output for the following two lines.



The list is restored to its initial state before each line executes:

```
Line1: printf("%d", head->next->next->next->prev->prev->data);
Line2: printf("%d", tail->prev->prev->prev->prev->next->next->data);
```

Which of the options is correct.

A. Output of line 1 is 7 and Output of line 2 is 9
B. Output of line 1 is 9 and Output of line 2 is 7
C. Both lines output 7
D. Both lines output 9

Your Answer: A | Correct Answer: A | Correct | Discuss

Q #2 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | DS

What is the run-time complexity of inserting a new element at the beginning of a circular, doubly-linked list with a head?

A. $O(1)$
B. $O(\log \mathrm{N})$
C. $O(\mathrm{N})$
D. $O(\mathrm{N}^2)$

Your Answer: A | Correct Answer: A | Correct | Discuss

Q #3 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | DS

Given a circular, doubly-linked list whose contents are sorted in ascending order, what is the run-time complexity for inserting a new element into the list so that it remains correctly sorted? (Including the time required to search for the element's correct position.)

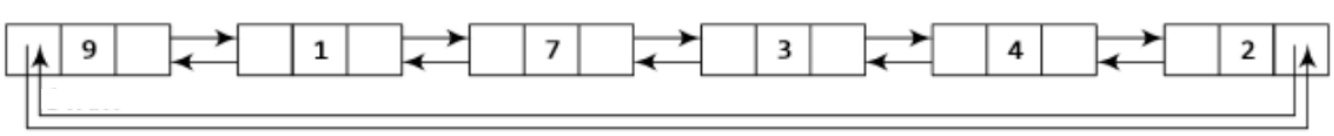
A. $O(1)$
B. $O(\log \mathrm{N})$
C. $O(N)$
D. $O(\mathrm{N}^2)$

Your Answer: C | Correct Answer: C | Correct | Discuss

Q #4 | Multiple Choice Type | Award: 1 | Penalty: 0.33 | DS

Consider a function findCLL that takes a doubly circular linked list head and an integer value as input.

One example of doubly circular linked list is given below –




```
    int findCLL(struct node * first, int n) {

        while (first -> value != n)
            first = first -> next;
 5.
        if (first -> value == n)
            return 1;
        else
            return -1;
10. }
```

Consider two statements $\mathrm{S1}$ and $\mathrm{S2}$ given below.

- $\mathrm{S1}$: Function returns 1 if there exists a value in linked list
- $\mathrm{S2}$: Function returns -1 if value does not exist in linked list

Which of the following is the correct option.

A. $\mathrm{S1}$ is True but $\mathrm{S2}$ is False.

B. $S2$ is True but $S1$ is False.
C. Both are True.
D. Both are False.

Your Answer: A **Correct Answer: A** Correct Discuss

Q #5 Multiple Choice Type | Award: 1 | Penalty: 0.33 | DS

Consider the following Doubly Linked List:

If head points to the first node of the linked list then what will be the output of the following node?

```
head=head->next->next->next->prev;
head->next->next->prev=head;
printf("%d",head->next->next->prev->next->value);
```

A. 1
B. 3
C. 5
D. 7

Your Answer: D **Correct Answer: D** Correct Discuss

Q #6 Multiple Choice Type | Award: 2 | Penalty: 0.67 | DS

```
    struct node{
        int data;
        struct node *next;
    }
 5. void print(struct node *ptr)
    {
        if(ptr)
        {
            printf("%d ",ptr->data);
10.         do {
                printf("%d ",ptr->data);
            }
            while(ptr->next);
        }
15. }
```

What is the output, if the address of the first node of singly linked list $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5$ is passed in the above C code?

A. $1\ 2\ 3\ 4\ 5$
B. $1\ 1\ 2\ 3\ 4\ 5$
C. $1\ 1\ 2\ 3\ 4\ 5\ 5$
D. None of these

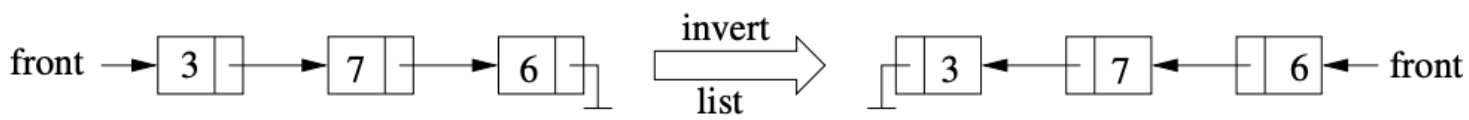
Your Answer: D **Correct Answer: D** Correct Discuss

Q #7 Multiple Choice Type | Award: 2 | Penalty: 0.67 | DS

Consider the following code fragment.

```
    struct node {
        int data;
        struct node *next;
    };
 5.     struct node *curr, *prev, *next;
        //front is pointing to the head of the linked list as shown
           in figure.
        curr = front;
        next = curr->next;
10.     prev = NULL;
        while (curr != NULL) {
        (*)
    }
    front = prev;
```

Which code must be added in the part marked (*) so the above code correctly inverts a non-empty singly linked list? See the figure to understand what "invert" means.



A.
```
    next->next = prev; prev = curr; curr = next;
    if (next != NULL) next = next->next;
```

B.
```
    curr->next= prev; prev = curr; curr = next;
    if (next != NULL) next = next->next;
```

C.
```
    next->next=curr; prev = curr; curr = next;
    if (next != NULL) next = next->next;
```

D.
```
    prev = curr; curr = next; curr->next= prev;
    if (next != NULL) next = next->next;
```

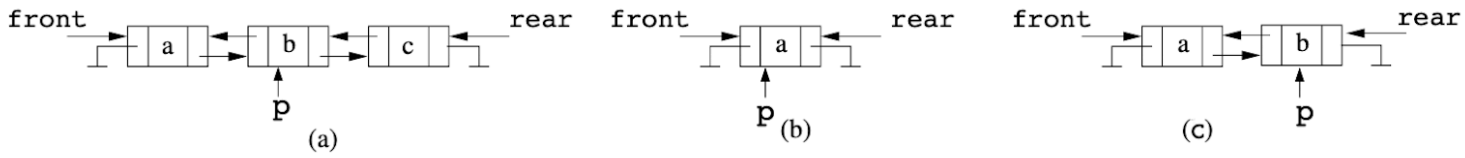
Your Answer: B | Correct Answer: B | Correct | Discuss

Q #8 | Multiple Select Type | Award: 2 | Penalty: 0 | DS

The following code is intended to remove a node p from a doubly linked list. Assume that we know that p is in the list, so the list is not empty.

```
    struct node {
        char value;
        struct node *next;
        struct node *prev;
 5. };
    struct node *prev *succ;
    prev = p->prev;
    succ = p->next;
    if (p == front) {
10.     front = front->next;
        if (front == NULL) rear = NULL;
        else front->prev = NULL;
    }
    else prev->next = succ;
15. if (p == rear) {
        rear = rear->next;
        rear->next = NULL;
    }
    else succ->prev = prev;
```

Given three linked lists (a),(b), and (c) as shown below.



Mark the correct option(s).

A. Code will crash on linked list (a)
B. Code will crash on linked list (b)
C. Code will crash on linked list (c)
D. Code will not crash on any of the given linked lists

Your Answer: B;C **Correct Answer: B;C** Correct Discuss

Q #9 Multiple Select Type Award: 2 Penalty: 0 DS

Consider a mutual pair of recursive functions g() and h().

```
    int g(struct node *l) {
        if (l == NULL || l->next == NULL) return 1;
        if (l->value < l->next->value) return h(l->next);
        else return 0;
 5. }

    int h(struct node *l) {
        if (l == NULL || l->next == NULL) return 1;
        if (l->value > l->next->value) return g(l->next);
10.     else return 0;
    }
```

Let head be a pointer to a singly linked list having at least 3 nodes.

When will the expression (g(head) || h(head)) return 1 or 0?

A. g(head) || h(head) is 1 if the linked list is in ascending order.
B. g(head) || h(head) is 1 if the linked list is in descending order.
C. g(head) || h(head) is 1 for every unsorted linked list.
D. g(head) || h(head) is 1 for the linked list $1 \rightarrow 3 \rightarrow 2 \rightarrow 4 \rightarrow 0 \rightarrow 6$

Your Answer: **Correct Answer: D** Not Attempted Discuss

Q #10 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | DS

Consider the following function that takes reference to head of a Doubly Linked List as parameter. Assume that a node of doubly linked list has previous pointer as $prev$ and next pointer as $next$.

```
    void fun(struct node **head_ref)
    {
        struct node *temp=NULL;
        struct node *current=*head_ref;
 5.     while(current!=NULL)
        {
            temp=current->prev;
            current->prev=current->next;
            current->next=temp;
10.         current=current->prev;
        }
        if(temp!=NULL)
            *head_ref=temp->prev;
    }
```

Assume that reference of head of following doubly linked list is passed to above function $1 \leftrightarrow 2 \leftrightarrow 3 \leftrightarrow 4 \leftrightarrow 5 \leftrightarrow 6$. What should be the modified linked list after the function call?

A. $2 \leftrightarrow 1 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 6 \leftrightarrow 5$
B. $5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1 \leftrightarrow 6$
C. $6 \leftrightarrow 5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 2 \leftrightarrow 1$
D. $6 \leftrightarrow 5 \leftrightarrow 4 \leftrightarrow 3 \leftrightarrow 1 \leftrightarrow 2$

Your Answer: | Correct Answer: C | Not Attempted | Discuss

Q #11 | Multiple Select Type | Award: 2 | Penalty: 0 | DS

Consider the following function Merge() that takes the head of two linked lists.

```
    struct node {
        int value;
        struct node *next;
    };
 5. typedef struct node Node;

    Node * Merge(Node * head1, Node * head2) {
        if (head1 == NULL) return head2;
        if (head2 == NULL) return head1;
10.     Node * head = NULL;
        if (head1 -> value < head2 -> value) {
            head = head1;
            head -> next = Merge(head1 -> next, head2);
        }
15.     else if (head1 -> value > head2 -> value) {
            head = head2;
            head -> next = Merge(head1, head2 -> next);
        }
    return head;
20. }
```

Assume that the input lists are correctly sorted. Which of the following are some of the possible behaviors when Merge() is executed with well-formed and valid inputs? Correctly merged linked list is merged sorted linked list.

A. The function will produce a correctly merged linked list.
B. The function may lead to a null pointer dereference.
C. The function may result in an incorrectly merged linked list.
D. Merge() will work on the following two lists.

- $\text{List1} : 1 \rightarrow 3 \rightarrow 5 \rightarrow 7 \rightarrow \text{Null}$
- $\text{List2} : 2 \rightarrow 4 \rightarrow 6 \rightarrow 8 \rightarrow 10 \rightarrow 12 \rightarrow \text{Null}$

Your Answer: C;D **Correct Answer: C;D** Correct Discuss

Q #12 Multiple Select Type Award: 2 Penalty: 0 DS

Consider the following program. printlist() is a function that takes the head of a linked list and prints all nodes values separated by comma. Node is typedefed singly linked list type struct.

```
    void insert1(Node *head,int data)
    {
        Node *NewNode= (Node *)malloc(sizeof(Node));
        NewNode->value=data;
 5.     NewNode->next=head;
        head=NewNode;
    }
    void insert2(Node **head_ref,int data)
    {
10.     Node *NewNode= (Node *)malloc(sizeof(Node));
        NewNode->value=data;
        NewNode->next=*(head_ref);
        *(head_ref)=NewNode;
    }
15. int main()
    {
        /* create a linked list 1->2->3->4->5
    and head points to the first node.*/
        insert1(head,9);
20.     printlist(head); //Line X

        //The list is restored to its initial state

        insert2(&head,9);
25.     printlist(head); //Line Y
    }
```

Which of the following is/are true about the above program?

A. Line X prints $9, 1, 2, 3, 4, 5$
B. Line Y prints $9, 1, 2, 3, 4, 5$
C. Line X prints $1, 2, 3, 4, 5$
D. Line Y prints $1, 2, 3, 4, 5$

Your Answer: **Correct Answer: B;C** Not Attempted Discuss

Q #13 Multiple Choice Type Award: 2 Penalty: 0.67 DS

Consider the following function fun() that takes the head of a linked list.

```
    struct node {
        int value;
        struct node *next;
    };
 5. typedef struct node Node;

    int fun(Node *head){
        if(head== NULL) return 1;
        Node *p,*q;
10.     p = head;
        q = p->next;
        while(q!=NULL && q!=p){
            q = q->next;
        if(q==NULL) return 1;
15.     q = q->next;
        p = p->next;
        }
    return (q==NULL);
    }
```

We say, a linked list has a loop if the last node of linked list points to some node of linked list and does not point to NULL.

What does the above function do?

A. Returns 0 is there is loop in linked list
B. Returns 1 is there is loop in linked list
C. Returns 0 is length of the linked list is even
D. Function may go to infinite loop if there is a loop in linked list

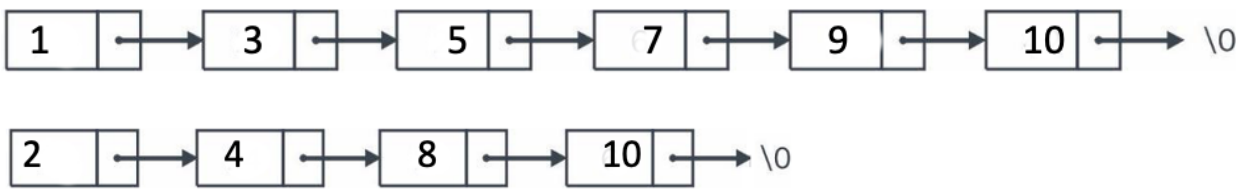
Your Answer: A **Correct Answer: A** Correct Discuss

Q #14 Multiple Choice Type Award: 2 Penalty: 0.67 DS

Consider the following function foo() which takes the head pointer of two singly-linked lists.

```
    struct node *foo(struct node *head1, struct node *head2)
    {
        struct node *final, *temp;
        if (head1 == NULL) return head2;
 5.     if (head2 == NULL) return head1;
        temp = foo(head1->next, head2->next);
        final = head1;
        head1 -> next = head2;
        head2 -> next = temp;
10.     return final;
    }
```

What will be the final linked list returned by foo() if executed upon following linked lists?



A. $1, 2, 3, 4, 5, 7, 8, 9, 10$
B. $1, 2, 3, 4, 5, 7, 8, 10, 9, 10$
C. $1, 2, 3, 4, 5, 8, 7, 10, 9, 10$
D. None of these

Your Answer: C **Correct Answer: C** Correct Discuss

Q #15 | Multiple Choice Type | Award: 2 | Penalty: 0.67 | DS

A doubly linked list is declared as:

```
    struct Node {
        int Value;
        struct Node *prev;
        struct Node *next;
5. };
```

Which of the following segment of code deletes the node pointed to by X from the doubly linked list, if it is assumed that X points to neither the first nor the last node of the list?

A. $\mathrm{X} \rightarrow$ prev $\rightarrow$ next $= \mathrm{X} \rightarrow$ next ; $\mathrm{X} \rightarrow$ next $\rightarrow$ prev $= \mathrm{X} \rightarrow$ prev; free(X);
B. $\mathrm{X} \rightarrow$ prev.next $= \mathrm{X} \rightarrow$ next ; $X.$ next $\rightarrow$ prev $= \mathrm{X} \rightarrow$ prev; free(X);
C. $\mathrm{X} \rightarrow$. prev $\rightarrow$ next $= \mathrm{X}$.prev ; $\mathrm{X} \rightarrow$ next.prev $= \mathrm{X}$.prev; free(X);
D. $\mathrm{X} \rightarrow$ prev $\rightarrow$ next $= \mathrm{X} \rightarrow$ prev ; $\mathrm{X} \rightarrow$ next $\rightarrow$ prev $= \mathrm{X} \rightarrow$ next; free(X);

Your Answer: A | **Correct Answer: A** | Correct | Discuss

You're doing Great!