Question Nos.		Marks Awarded

Q1. Big O
Big theta
Big Ω
Big O gives us the upper limit of the case the upper limit (Worst Case)
Big theta gives us the lower bound of the case (Best Case)
Big Ω gives us the combination of both Big O and Big Ω
Big (O) notations helps us to understand the worst case of a given algorithm it is generally represented as.
$O(1) \rightarrow O(\log n) \rightarrow O(n) \rightarrow O(n \log n)$.
$\rightarrow O(n^2) \rightarrow O(n^{\log n})$
where $O(1)$ is best.

Question Nos.

Marks Awarded

Q2

1. Insertion at the start

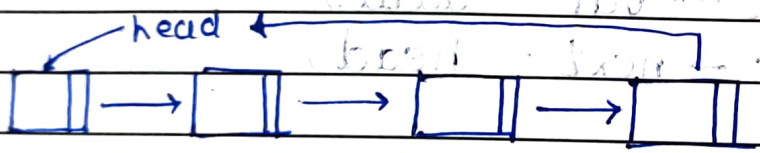
```
struct node {
    int val;
    struct node *next;
};

function InsertatStart (struct node ** head, data
{
    struct node * newnode;
    newnode → val = data;
    newnode → next = head;
    struct node temp = head
    while ( temp → next != head )
        temp = temp → next
    temp → next = newnode;
}
```

```
function DeletionOfLastnode (struct node ** head, d
{ struct node prev
    temp;
    struct node * newnode = head.
    while ( temp → next != head )
        temp = temp → next
    prev → next = prev → next
        prev = temp
    prev → next = head.
```

Question Nos | Marks Awarded

free(temp)

function DeletionOfLastNode(struct node **head)
{
struct node prev;
struct node temp = head.
while(temp→next != head)
prev→next = temp.
temp ⇒ temp→next;
prev→next = head.
free(temp)
}.



above is a circular link list in this the end node contains the memory location of the head instead of NULL pointer so we can use this to traverse to the last node until the next points to the head.

Insertion at start
create a newnode then traverse to the end node and set the newnode as the pointer to the last node

Deletion of last node
traverse to end store the previous node free the last node and set previous node to head.

Question Nos.

Marks Awarded

Q3. (A + B * (C - D) ^ E) / (F - G + H * I)

var	stack	Output
(	(	
A	(	A
+	(+	A
B	(+	AB
*	(+ *	AB
(	(+ * (	AB
C	(+ * (	ABC
-	(+ * (-	ABC
D	(+ * (-	ABCD
)	(+ * (-) — pop	ABCD-
^	(+ * ^	ABCD-
E	(+ * ^	ABCD-E
)	(+ * ^) — pop	ABCD-E^
/	/	ABCD-E^*+
(	/ (	
F	/ (	ABCD-E^*+F
-	/ (-	
G	/ (- — pop	ABCD-E^*+FG
+	/ (+	ABCD-E^*+FG-
H	/ (+	ABCD-E^*+FG-H
*	/ (+ *	
I	/ (+ *	ABCD-E^*+FG-HI
)	/ (+ *) — pop	ABCD-E^*+FG-HI*
		ABCD-E^*+FG-HI*+/

Question Nos

Mar Awar

Infix to POSTFIX

ABCD-E^*+FG-HI*+/