Question Nos. | Marks Awarded

| | | |
|---|---|---|
| / | / | ABCD-E^*+ |
| ( | /( | ABCD-E^*+ |
| F | /( | ABCD-E^*+F |
| - | /(- | ABCD-E^*+F |
| G | /(- | ABCD-E^*+FG |
| + | /(+ | ABCD-E^*+FG- |
| H | /(+ | ABCD-E^*+FG-H |
| * | /(+* | ABCD-E^*+FG-H |
| I | /(+* | ABCD-E^*+FG-HI |
| ) | / | ABCD-E^*+FG-HI*+ |
| | | ABCD-E^*+FG-HI*+/ |

∴ The final postfix expression is :

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

Q2] assuming that a circular linked list has ~~also~~ already been created with a head pointer pointing to the first node.

1. Insertion at the start

```
Algo Insert_beg {
IF(AVAIL == NULL)
    Write "SPACE UNAVAILABLE"
ELSE
    NEWNODE = AVAIL
NEWNODE → DATA = NEWDATA
NEWNODE → NEXT = HEAD;
```

Question Nos. | Marks Awarded

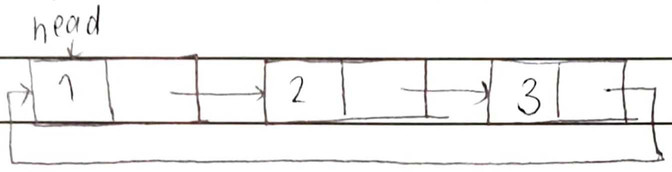
```
WHILE (ptr -> ~~next~~ NEXT != ~~head~~ HEAD)
    ptr = ptr -> ~~next~~ NEXT
ptr -> next = NEWNODE
HEAD = NEWNODE
}
```

code in C:

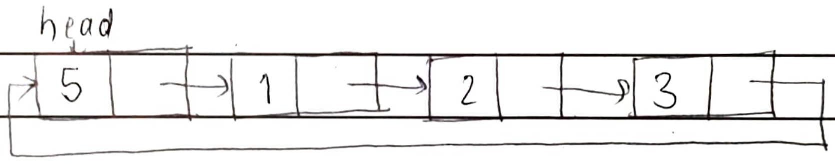
```
void insert_beg (struct node *head)
{
    struct node *ptr = head;
    int val;
    Struct node *newnode = NULL;
    newnode = (struct node *) malloc (sizeof (struct node));
    ~~newnode -> data = val;~~
    ~~newnode -> next = head;~~
    if (newnode == NULL)
    {
        printf ("Memory allocation failed ");
        return;
    }
    newnode -> data = val;
    newnode -> next = head;
    while (ptr -> next != head)
    {
        ptr = ptr -> next;
    }
    if (ptr = NULL)
    {
        printf ("Not a circular linked list);
        }
        return;
    }
    ptr -> next = newnode;
    head = newnode;
}
```

~~2. Deletion of last node~~

Example



After insertion :



2. Deletion of last node

```
Algo ~~sas~~ DELETE_End()
PTR = HEAD
PREPTR = HEAD
WHILE(PTR -> NEXT != HEAD)
{
    PREPTR = PTR
    PTR = PTR -> NEXT
}
IF PTR == NULL
    Write "Not a Circular Linked List"
    Go to ~~step~~ END
```

Question Nos.

Marks Awarded

```
FREE PTR
PREPTR → NEXT = HEAD
END
}
```

code in C:

```
void delete_last (struct node *head)
{
    struct node *preptr = head;
    struct node *ptr = head;

    while (ptr → next != head)
    {
        preptr = ptr;
        ptr = ptr → next;
    }
    if (ptr == NULL)
    {
        printf ("Not a circular linked list");
        return;
    }
    free (ptr);
    preptr → next = head;
}
```

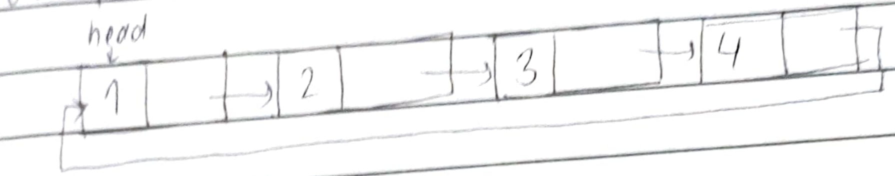
[illegible]

NEXT PAGE →

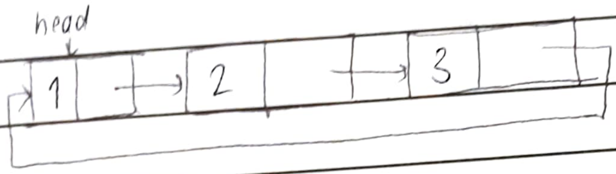
Question Nos

Marks Awarded

Example



After deletion :



For insertion at the start, a new node was created using malloc. Newnode points to head and the last node points to the newnode. After this, we shift head to the new node. This ensures that the new node becomes the first node of the circular linked list.

For deletion of the last node, we make 2 pointers. ptr points to the last node while preptr points to the node before ~~preptr~~. the ~~th~~ last node. We use free() to delete the last node and the ~~seco~~ new last node (prepTr) now points to head.

Q1] Asymptotic notations are used to represent time complexity of algorithms. They are as follows:
(i) Big-Oh (O) - Upper bound
This represents the worse-case time complexity. Thus, this is the highest amount of time required