O/P:-

Enter a valid parenthesized infix expression:-
(A+(B*C-(D/E^F)*G)*H)

- Postfix expression: ABC*DEF^/G*-H*+

B

13/10/25 lab pgm-3

a) WAP to simulate the working of queue of integers using an array. Provide the following operations: Insert, Delete, Display. The pgm should print appropriate messages for queue empty & overflow conditions

Pseudocode

```
// initialize or declare array
int queue[5];
int front=-1; rear=-1;
void enque(){
// insertion
int x;
if (rear=4) => Queue overflow
else if front=rear=-1 => front=rear=0, queue[rear]=x
                                    // insert integer
  else => rear++;
       queue[rear]=x;
}
void deque(){
  if front=rear=-1 => print "Underflow"
  else if (front==rear) => front=rear=-1;
  else => printf("Deleted element",
          front++
}
void display(){ ---> if front=rear=-1 => stack empty
                <=rear
  for (i=front; i<=0; i++) {
    printf("%d", queue[i]);
}
```

```
void main(){
    int ch; int y;
do{
    printf("Enter element:");
    scanf("%d", &ch);
  switch(ch){
    case 1:
        enque
        printf("enter element to insert:");
        scanf("%d", &y);
        enque(y);
        break;
    case 2:
        deque();
        break;
    case 3:
        display();
        break;
    case 4:
        break;
    default:
    }
  while(ch!=4);
```

code

```
#include<stdio.h>
#define N 5
int queue[N];
int front=-1; rear=-1;
void enque(int x){
    if (rear==(N-1)) {
        Printf("Queue overflow\n");
} else if ((front==-1)&&(rear==-1)){
        front=rear=0;
        queue[rear]=x;
  } else{ rear++;
}}      queue[rear]=x;
```

```
void display(){
    if((front==-1)&&(rear==-1)){
        printf("The Queue is empty\n");
    }else{
        for(i=front;i<=rear;i++){
            printf("%d",queue[i]);
        }
    }
}
void dequeue(){
    if((front==-1)&&(rear!=-1)){
        Printf("The Queue is Empty\n");
    }else if(front==rear){
        printf("Deleted element is: %d",queue[front]);
        front=rear=-1;
    }else{
        Printf("Deleted element is %d",queue[front]);
        front++;
    }
}
void main(){
    int ch,y;
    do{
        printf("\n Enter the element you want to
                insert:");
        scanf("%d",&y);
        enqueue(y);
        break;
        switch(ch){
            case1:
```

```
Void main(){
    int ch,y;
    printf("Enter your choice between 1 to 4:");
    scanf("%d",&ch);
    switch(ch){
        case 1:
            printf("\nEnter the element you want
                to insert:");
            scanf("%d",&y);
            enque(y);
            break;
        case 2:
            deque();
            break;
        case 3:
            display();
            break;
        case 4:
            break;
        default:
            printf("your choice is invalid!");

    }
    }while(ch!=4);

}
```

op

enter your choice :2

The Queue is empty

Enter your choice:3

The queue is empty

enter your choice: 1
Enter the element: 11
Enter your choice: 1
Enter the element: 12
Enter your choice: 1
Enter the element: 13
Enter your choice: 1
Enter the element: 14
Enter your choice: 1
Enter the element: 15
Enter your choice: 1
Enter the element: 16
Queue overflow
Enter your choice: 3
11 12 13 14 15
Enter your choice: 2
Deleted element is: 11
Enter your choice: 4

13/6/25

3/7/25

Q) WAP to simulate the working of a circul
of integers using an array. Provide the fo
operations, Insert, Delete & Display. The progr
Print appropriate messages for queue emp
queue overflow conditions

Pseudo code

void enqueue(int x){