Name _____ Std _____ Sec _____

Roll No. _____ Subject _____ School/College _____

School/College Tel. No. _____ Parents Tel. No. _____

Sl. No.	Date	Title	Page No.	Teacher Sign / Remarks
1.	29/09/25	Push, pop, peek		10
2.	06/10/25	Infix to postfix		10 6/10/25
~~3~~	~~13/11/25~~	~~Insert, delete, disp~~		
3.	13/11/25	Insert, delete, display		10 13/10

Date __/__/__
Page _____

LAB PROGRAM-3

W.A.P. to Simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display the program should print appropriate messages for queue. Empty & queue overflow conditions.

Algorithm:

INITIALISE

- Set front → -1
- Set rear → -1
- Define array queue [MAX]

INSERT

- If rear = MAX - 1
  Print ("Queue overflow")
  Exit from function
- If front = -1
  Set front = 0
- Set rear = rear + 1
- Get queue [rear] → item
- Print ("item inserted")

DELETE

- If front = -1 && rear = -1
  Print ("queue is empty")
  Exit from function
- ~~Set item → queue [front]~~
- If front = rear
  Set front = rear = -1

Date __/__/__
Page

- If front > rear
  Set front++
  Print (item)

DISPLAY

- If front = -1.
  Print ("queue is empty")
  Exit from function

- for i from front to rear
  Print (queue[i])

MAIN

- Repeat until user exits.
- Print insert, delete, display.
- Run switch case for all cases and then call the functions.

Code:

```
#include<stdio.h>
#include<stlib.h>
#define N 5

int front = -1;
int rear = -1;
int queue[N];

void enqueue(int x){
    if(rear == N-1){
        printf("queue overflow");
    }
```

```
else if ( front == -1 && rear == -1) {
    front = rear = 0;
    queue[rear] = x;
  }
}

void dequeue() {
    if (front == -1 && rear == -1) {
      printf("queue is empty");
    }
  else if (front == rear) {
      front = rear = -1;
    }
  else {
      printf("deleted element = %d \n", queue[front]);
      front ++;
    }
}

void display() {
    int i;
  for(int i = front; i < rear++ ; i++) {
      printf("%d \n", queue[i]);
  }
}

int main() {
    int ch;
  printf("1. Insert \n 2. Delete \n 3. Display \n 4. Exit \n");

while (1) {
    int x;
  Printf("enter choice \n");
  scanf("%d", &ch);
```

```
switch(ch){
    Case 1: printf("Enter the number:");
            scanf("%d", &x);
            enqueue(x);
            break;
    Case 2: dequeue();
            break;
    Case 3: display();
            break;
    Case 4: return 0;

    default: printf("Invalid choice");
    }
}
return 0;
}
```

Output:

1. Insert
2. Delete
3. Display
4. Exit

Enter Choice: 2

~~Queue is empty~~ Queue is empty

Enter choice: 1

Enter the number: 23

Element inserted

Enter Choice: 1

Enter the number: 56

Element inserted

Enter choice: 1

Enter the number: 89

Date __/__/__
Page _____

Element inserted
Enter choice: 1
Enter the number: 74
Element Inserted
Enter choice: 1
Enter the number: 58
element inserted
Enter choice: 1
Enter the number: 12
Queue overflow
Enter choice: 3
23
56
89
74
58
Enter choice: 2
Element deleted = 23
enter choice
3
56
89
74
58
enter choice
4

13/10