3/4/25

LAB PROGRAM - 3b

Q WAP to simulate the working of circular queue, of integers using array, provide the following operations;

Insert
delete
display

Program should print appropriate messages for empty queue, full queue, and queuefue conditions.

Pseudocode:

```
Set front = rear = -1 ;
Enque:
 If (front = -1 && rear = -1) {
         Set front = rear = 0 ;
         queue[rear] = x ;

 else if ((rear +1) % N = = front) {
         printf ("Queue is full ");

 else {
       rear = (rear+1) % N ;
       queue [rear] = x ;
 }
Deque:
 if ( front = -1 && rear = -1) {
         printf("Queue is empty");
 }
 else if ( front = = rear) {
      ~~set front~~
      printf ("deleted element : %d", queue [front]);
      front = front +1% N;
}
else { printf ("deleted element : %d", queue [front]);
}     front = front + 1% N ;
```

display

```
    for (int i = front ; i< = rear ; (i+1)%N ){
              printf ("queue is %d", queue [i]);
    }
```

3/11/25

Code:

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

```
void  deque (){
      if (front = = -1 && rear = = -1){
           printf ("queue   underflow \n");

      else if ( front = =rear){
             printf ("deque = %d \n", queue [front]);
             front = rear = -1;
      }
      else {
           printf ("deque = %d \n", queue [ front]);
           front = (front +1)%n;

      }
}

void  display (){
      for (int i = front ; ; i= (i+1)% n){
           printf ("%d \t", queue[i]);
           if (i= = rear){
                 break;
           }
           printf ("\n");
```

8
3/11
o/p
[illegible]

```
int main( ){
      int ch;
      printf (" 1. enque \n 2. deque \n 3. display\n");
      while (1) {
            int x;
            printf (" enter choice : ");
            scanf (" %d ", &ch);

            switch (ch){
                  case 1 : printf ("enter number to insert : ");
                           scanf (" %d", &x);
```

```
                    enque(x);
                    break;
            case 2 : deque();
                     break;
            case 3 : display();
                     break;
            case 4 : return 0;
            default : printf("Invalid choice");
            }
    }
    return 0;
}
```

O/P:

```
1. enque
2. deque
3. display
enter choice : 2
Queue Underflow
enter choice : 1
enter number to insert : 10
enter choice : 1
Enter number to insert : 20
enter choice : 1
enter number to insert : 30
enter choice : 1
enter number to insert : 40
Enter choice : 1
enter number to insert : 50
enter choice : 1
enter number to insert : 60
Queue overflow
enter choice : 2
deque = 10
enter choice : 2
deque = 20
enter choice : 3
30  40  50
enter choice : 4
```