8/1/24

1> Write a program to simulate the working of the queue of integers using arrays. Provide the following operations: enqueue, dequeue, display.

→
```
# include <stdio.h>
# include <math.h>
# include <string.h>
# define N 5
  int queue [N];
int front = -1;
int rear = -1 ;

void enqueue (int n) {

    if (rear == N-1) {
      printf ("overflow");
    }
else if (front == -1 && rear == -1) {
        front = rear = 0
        queue [rear] = x;
    }
 else {
        rear++;
        queue [rear] = x;
      }
 }

void dequeue () {
    if (front = = -1) {
        printf ("Underflow");
    }
 else if (front = = rear) {
      printf (" The dequeued element
      is %d", queue [front]);
     front = rear = -1;
  }
```

```
else {
printf (" The dequeued element is %d",
                    queue [front]);
       }
  }

void display () {

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

 int main () {
        int ch, x;
while (ch! = 0) { printf ("Enter 1:
enqueue /n  2: dequeue \n  3: display\n
4: terminate program");
        scanf (" %d", &ch);
```

```
switch (ch) {
case 1: printf ("Enter value:")
        scanf ("%d", & x);
        enqueue (x);
        break;

case 2: dequeue ();
        break;

case 3: display ();
        break;

case 4: printf ("Terminating program");
         break;
default: printf ("Invalid input");
                                break;
        }
  }
return 0;
}
```

Page No.
Date / /

Q2> Write a program to simulate the working of a circular queue using array. Provide the following operation insert, delete & display. The program should print appropriate message for queue empty and queue overflow condition

→
```
# include <stdio.h>
# include <math.h>
# include <string.h>
  # define N 7
    int queue [N];
   int front = -1;
   int rear = -1;

    void enqueue (int x) {
```

```
if ( front == -1 && rear == -1) {
        front = rear = 0;
        queue [rear] = x;
    }
else if (((rear+1) % N) == front) {
        printf (" overflow ");
    }
else {
        rear ++;
        queue [rear] = x;
    }


void dequeue () {

        if (front == -1 && rear == -1) {
        printf (" underflow ");
        }
    else if (((front + 1) % N) == rear) {
```

```
printf ("The dequeued element is %d"
                    queue [front]);
      front = rear = -1;
    }
  else { printf (" The dequeued element
is %d", queue [front] front = (front +1)
% N;
  }
}

void display () {

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

```
int main() {
  int ch, x;
while (ch != 0) {
printf ("Enter 1: enqueue/n
 2: dequeue /n   3: display /n
 4: terminate program ");
scanf ("%d", & ch);

switch (ch) {

case 1:  printf ("Enter value : ");
         scanf ("%d", & x);
         enqueue (x)
         break;
case 2:  dequeue ();
         break;
case 3:  display ();
         break;}
case 0: printf("Terminating program");
        break;
default: printf("Invalid input");
        break;
   }
}
return 0;
 }.
```