1) Write a program to simulate the working of the queue of integers using an array. provide the following operations : Insert, delete, display. The program should print appropriate message for overflow and underflow condition.

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

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

void dequeue () {
        if (front == -1) {
          printf ("underflow");
```

```
    }
    else if (front == rear) {
        front = rear = -1;
    }
    else { front++;
    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
        3: display \n 4: terminate program"
    scanf("%d", &ch);
switch(ch) {
    case1: printf("Enter value")
            scanf("%d", &x)
            enqueue(x);
            break;
    case 2: dequeue();
            break;
    case 3: display();
            break;
```

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

2> Write a program to simulate the working of a circular queue using an array. Provide the following operations: insert, delete & display. The program should print appropriate messages for queue empty and queue overflow condition.

→

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

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

```
else {
    rear++;
    queue[rear] = x;
}

void dequeue() {
    if (front == -1 && rear == -1) {
        printf ("underflow");
    }
    else if ((front+1)%N) == rear) {
        front = rear = -1;
    }
    else { printf ("The dequeue 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;
default case 0 : printf (" Terminating
        program ");
        break;
default : printf (" Invalid input ");
        break;
    }
}
return 0;
}
```

OUTPUT :-

① Enter 1 : enqueue
2 : Dequeue
3 : display
0 : Terminate program

enqueue (2)
enqueue (5)
dequeue ()
enqueue (8)
enqueue (9)
enqueue (10)
enqueue (12) ← overflow
display () → 5 8 9 10

(2) Enter 1: enqueue
2: dequeue
3: display
0: Terminate program
enqueue (15)
enqueue (20)
enqueue (30)
dequeue ()
display () ← 20 30

Jun 8/12/24