8/1/24

1) Write a program to simulate the working of the queue of integers using array. 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 x){

    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){
        front
        printf("The dequeued element is %d",
                queue[front]);
        front = rear = -1;
    }
    else 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 \n
                3: display \n 4: terminate program");
    scanf("%d",&ch);
```

```
while (ch != 0) {
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;
}
```

2)

→

o/p ⇒ Enter
1: Enqueue
2: Dequeue
3: Display
0: Terminate program:
1
Enter Value: 56

2) 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("\nUnderflow");
    }
    else if(((front+1)%N)==rear){
        front=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;
while(ch!=0){
    switch(ch){
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;
}
```

① o/p

Enter:
1: Enqueue
2: Dequeue
3: Display
0: Terminate program !
Enter value : 2
Enter:
1: Enqueue
2: Dequeue
3: Display
0: Terminate program !
Enter value : 3
Enter:
1: Enqueue
2: Dequeue
3: Display
0: Terminate program !
Enter value : 5
Enter:
1: Enqueue
2: Dequeue
3: Display
0: Terminate program !
Enter value : 8
Enter:
1: Enqueue
2: Dequeue
3: Display
0: Terminate program !
Enter value : 9

Display() => 2 3 5 8 9

Enter:

1: Enqueue

2: Dequeue

3: Display

0: Terminate program 1

Enter value: 55

Overflow

Enter:

1: Enqueue

2: Dequeue

3: Display

0: Terminate program 2

Enter:

1: Enqueue

2: Dequeue

3: Display

0: Terminate program 2

Dequeue (2)

Dequeue (3)

Dequeue (5)

Dequeue (8) ← Underflow condition

Sun
8/1/24