ASSIGNMENT - 7

①

```
int items[SIZE];
int front = -1, rear = -1;

int isFull() {
  if ((front == rear + 1) || (front == 0 && rear == SIZE - 1))
  return 1;
  return 0; }

int isEmpty() {
  if (front == -1) return 1;
  return 0; }

void enQueue(int element) {
  if (isFull())
  printf("Queue is full!\n");
  else {
    if (front == -1) front = 0;
    rear = (rear + 1) % SIZE;
    items[rear] = element;
    printf("Inserted -> %d", element); } }

int deQueue() {
    int element;
    if (isEmpty()) {
      printf("Queue is empty!!\n");
      return (-1); }
  else {
      element = items[front];
      if (front == rear) {
        front = -1;
        rear = -1; }
  else {
        front = (front+1) % SIZE; }
```

```
printf ("Deleted element -> %d \n", element);
return (element); }

void display() {
    int i;
    if (isEmpty())
        printf ("Empty Queue \n");
    else {
        printf ("Front -> %d", front);
        printf ("Items -> ");
        for (i= front; i!=rear; i=(i+1)%SIZE) {
            printf ("%d", items[i]); }
        printf ("%d", items[i]);
        printf ("Rear -> %d \n", rear); }}
```