```
#include <stdio.h>
#include <stdlib.h>
/*
WORKING INTEGER QUEUE
*/
#define MAX 6

void enqueue(int [], int *f,int *r,int val); //enter element in queue
int dequeue(int [], int *f,int *r); //remove element from queue
void display(int [], int f,int r); //display queue elements
int isFull(int, int);
int isEmpty(int, int);

int isFull(int f, int r){
  if ((f == 0 && r == MAX-1)||(f==r+1))
    return 1;
  return 0;
}
int isEmpty(int f, int r){
  if (f==-1 && r==-1)
    return 1;
  return 0;
}
void enqueue(int que[], int * front, int * rear, int val ){
  //check if full
  if (isFull(*front, *rear)){
    //queue is full
    printf("Queue overflow\n");
    return;
  }
  //if its empty
  if (*front == -1 && *rear == -1){
    *front = 0;
```

```
    *rear = 0;
    que[0] = val;
  }
  //as long as queue is less than MAX-1
  else if (*rear == MAX-1)
    *rear = 0;
  else
    (*rear)++;
  que[*rear] = val;
}
int dequeue(int que[], int * front, int * rear){
  //check if empty
  if (isEmpty(*front, *rear)){
    //printf("Empty queue\n");
    return -1;
  }
  int n;
  if (*front == *rear){
    n = que[*front];
    *front = -1;
    *rear = -1;
    return n;
  }
  n = que[*front];
  *front = *front + 1;
  return n;
}
void display(int que[], int f,int r){
  //printf("F: %d\tR:%d\n",f,r);
  if (isEmpty(f, r)){
    printf("Empty queue\n");
    return;
  }
```

```
  int i = f, j = r;
  if (i > j){
    while (i < MAX)
      printf("%d ", que[i++]);
    i = 0;
  }
  while (i <= j){
    printf("%d : ",i);
    printf("%d\n ", que[i++]);
  }
}
```