ENQUEUE&DEQUEUE

#include <stdio.h>

#include <stdlib.h>

#define MAX\_SIZE 10

typedef struct {

int arr[MAX\_SIZE];

int front, rear;

} Queue;

void initializeQueue(Queue \*q) {

q->front = -1;

q->rear = -1;

}

int isEmpty(Queue \*q) {

return (q->front == -1);

}

int isFull(Queue \*q) {

return (q->rear == MAX\_SIZE - 1);

}

void enqueue(Queue \*q, int value) {

if (isFull(q)) {

printf("Queue is full. Cannot enqueue.\n");

return;

}

if (isEmpty(q)) {

q->front = 0;

}

q->rear++;

q->arr[q->rear] = value;

}

void dequeue(Queue \*q) {

if (isEmpty(q)) {

printf("Queue is empty. Cannot dequeue.\n");

return;

}

printf("Dequeued element: %d\n", q->arr[q->front]);

if (q->front == q->rear) {

q->front = -1;

q->rear = -1;

} else {

q->front++;

}

}

void display(Queue \*q) {

if (isEmpty(q)) {

printf("Queue is empty.\n");

return;

}

printf("Queue elements: ");

for (int i = q->front; i <= q->rear; i++) {

printf("%d ", q->arr[i]);

}

printf("\n");

}

int main() {

Queue q;

initializeQueue(&q);

enqueue(&q, 10);

enqueue(&q, 20);

enqueue(&q, 30);

display(&q);

dequeue(&q);

display(&q);

return 0;

}