**// C Program to implement the circular queue in c using arrays**

#include <stdio.h>

// Define the maximum size of the queue

#define MAX\_SIZE 5

// Declare the queue array and front, rear variables

int queue[MAX\_SIZE];

int front = -1, rear = -1;

// Function to check if the queue is full

int isFull()

{

// If the next position is the front, the queue is full

return (rear + 1) % MAX\_SIZE == front;

}

// Function to check if the queue is empty

int isEmpty()

{

// If the front hasn't been set, the queue is empty

return front == -1;

}

// Function to enqueue (insert) an element

void enqueue(int data)

{

// If the queue is full, print an error message and

// return

if (isFull()) {

printf("Queue overflow\n");

return;

}

// If the queue is empty, set the front to the first

// position

if (front == -1) {

front = 0;

}

// Add the data to the queue and move the rear pointer

rear = (rear + 1) % MAX\_SIZE;

queue[rear] = data;

printf("Element %d inserted\n", data);

}

// Function to dequeue (remove) an element

int dequeue()

{

// If the queue is empty, print an error message and

// return -1

if (isEmpty()) {

printf("Queue underflow\n");

return -1;

}

// Get the data from the front of the queue

int data = queue[front];

// If the front and rear pointers are at the same

// position, reset them

if (front == rear) {

front = rear = -1;

}

else {

// Otherwise, move the front pointer to the next

// position

front = (front + 1) % MAX\_SIZE;

}

// Return the dequeued data

return data;

}

// Function to display the queue elements

void display()

{

// If the queue is empty, print a message and return

if (isEmpty()) {

printf("Queue is empty\n");

return;

}

// Print the elements in the queue

printf("Queue elements: ");

int i = front;

while (i != rear) {

printf("%d ", queue[i]);

i = (i + 1) % MAX\_SIZE;

}

// Print the last element

printf("%d\n", queue[rear]);

}

**// Main function**

int main()

{

// Enqueue some elements

enqueue(10);

enqueue(20);

enqueue(30);

// Display the queue

display();

// Dequeue an element and print it

printf("Dequeued element: %d\n", dequeue());

// Display the queue again

display();

**// End of main function**

return 0;

}

```