```c
#include <stdio.h>
#include <stdbool.h>

#define MAX_SIZE 100

typedef struct {
   int array[MAX_SIZE];
   int front;
   int rear;
} Queue;

void initializeQueue(Queue *queue) {
   queue->front = -1;
   queue->rear = -1;
}

bool isEmpty(Queue *queue) {
   return (queue->front == -1 && queue->rear == -1);
}

bool isFull(Queue *queue) {
   return (queue->rear + 1) % MAX_SIZE == queue->front;
}

void enqueue(Queue *queue, int value) {
   if (isFull(queue)) {
      printf("Queue is full, cannot enqueue.\n");
      return;
   } else if (isEmpty(queue)) {
      queue->front = queue->rear = 0;
   } else {
      queue->rear = (queue->rear + 1) % MAX_SIZE;
   }
   queue->array[queue->rear] = value;
}

int dequeue(Queue *queue) {
   int value;
   if (isEmpty(queue)) {
      printf("Queue is empty, cannot dequeue.\n");
      return -1;
   } else if (queue->front == queue->rear) {
      value = queue->array[queue->front];
      queue->front = queue->rear = -1;
   } else {
      value = queue->array[queue->front];
      queue->front = (queue->front + 1) % MAX_SIZE;
   }
   return value;
}

int main() {
   printf("K.R.Vishnu Chaithanya\n");
   printf("192372057\n");
   Queue queue;
   initializeQueue(&queue);
enqueue(&queue,50);
   enqueue(&queue, 20);
   enqueue(&queue, 30);
   printf("Dequeued element: %d\n", dequeue(&queue));
   printf("Dequeued element: %d\n", dequeue(&queue));
   printf("Dequeued element: %d\n", dequeue(&queue));
   printf("Dequeued element: %d\n", dequeue(&queue));
   return 0;
}
```

```
/tmp/IZND84xr3i.o
K.R.Vishnu Chaithanya
192372057
Dequeued element: 50
Dequeued element: 20
Dequeued element: 30
Queue is empty, cannot dequeue.
Dequeued element: -1
```