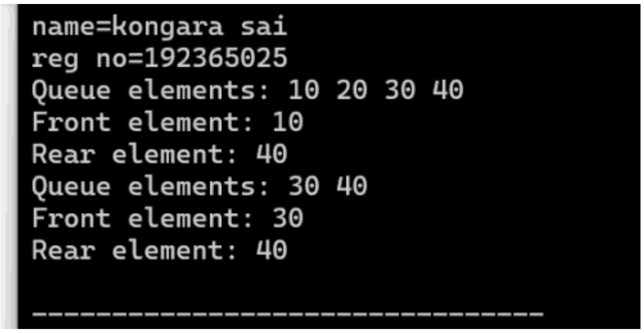
# Queue using array

Sunday, 10 March 2024 2:49 pm

```
#include <stdio.h>
#define MAX_SIZE 100

int queue[MAX_SIZE];
int front = -1, rear = -1;

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

 int isFull() {
    if (rear == MAX_SIZE - 1)
        return 1;
    else
        return 0;
}

 void Enqueue(int value) {
    if (isFull()) {
        printf("Queue is full. Cannot Enqueue.\n");
        return;
    } else if (isEmpty()) {
        front = rear = 0;
    } else {
        rear++;
    }
    queue[rear] = value;
}

 void Dequeue() {
    if (isEmpty()) {
        printf("Queue is empty. Cannot Dequeue.\n");
        return;
    } else if (front == rear) {
        front = rear = -1;
    } else {
        front++;
    }
}

 void Display() {
    if (isEmpty()) {
        printf("Queue is empty.\n");
        return;
    }

    printf("Queue elements: ");
    for (int i = front; i <= rear; i++) {
        printf("%d ", queue[i]);
    }
    printf("\n");
}

 int frontElement() {
    if (isEmpty()) {
        printf("Queue is empty.\n");
        return -1;
    }
    return queue[front];
}

 int rearElement() {
    if (isEmpty()) {
        printf("Queue is empty.\n");
        return -1;
    }
    return queue[rear];
}

int main() {

                                printf("name=kongara sai\nreg no=192365025\n");

    Enqueue(10);
    Enqueue(20);
    Enqueue(30);
    Enqueue(40);

    Display();

    printf("Front element: %d\n", frontElement());
    printf("Rear element: %d\n", rearElement());

    Dequeue();
    Dequeue();

    Display();

    printf("Front element: %d\n", frontElement());
    printf("Rear element: %d\n", rearElement());

    return 0;
}
```



```
name=kongara sai
reg no=192365025
Queue elements: 10 20 30 40
Front element: 10
Rear element: 40
Queue elements: 30 40
Front element: 30
Rear element: 40


--------------------------------
```