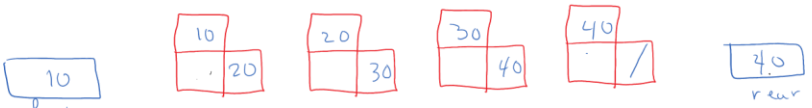
```
Struct node {
    int data
    node * next
}
node * front, * rear
```

Structure

```
enqueue (front, rear, data) ;
int data = dequeue (front, rear) ;
```

getting access to members

Front → data
Front → next



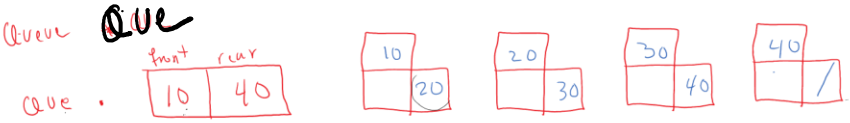
Structure inside another Structure

```
Struct Queue {
    node * front
    node * rear
}
Queue Que
```

```
enqueue (Que, data) ;
int data = dequeue (Que) ;
```

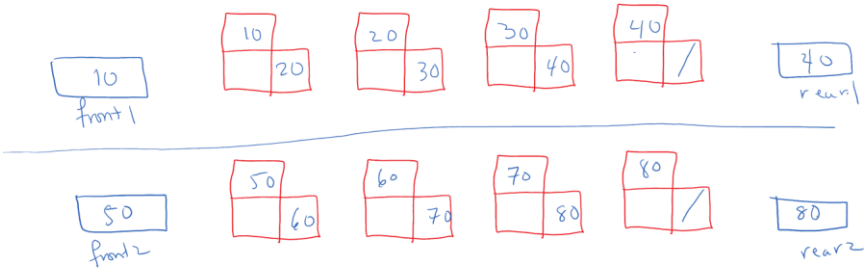
Que . Front → data
Que . Front → next

getting access to members



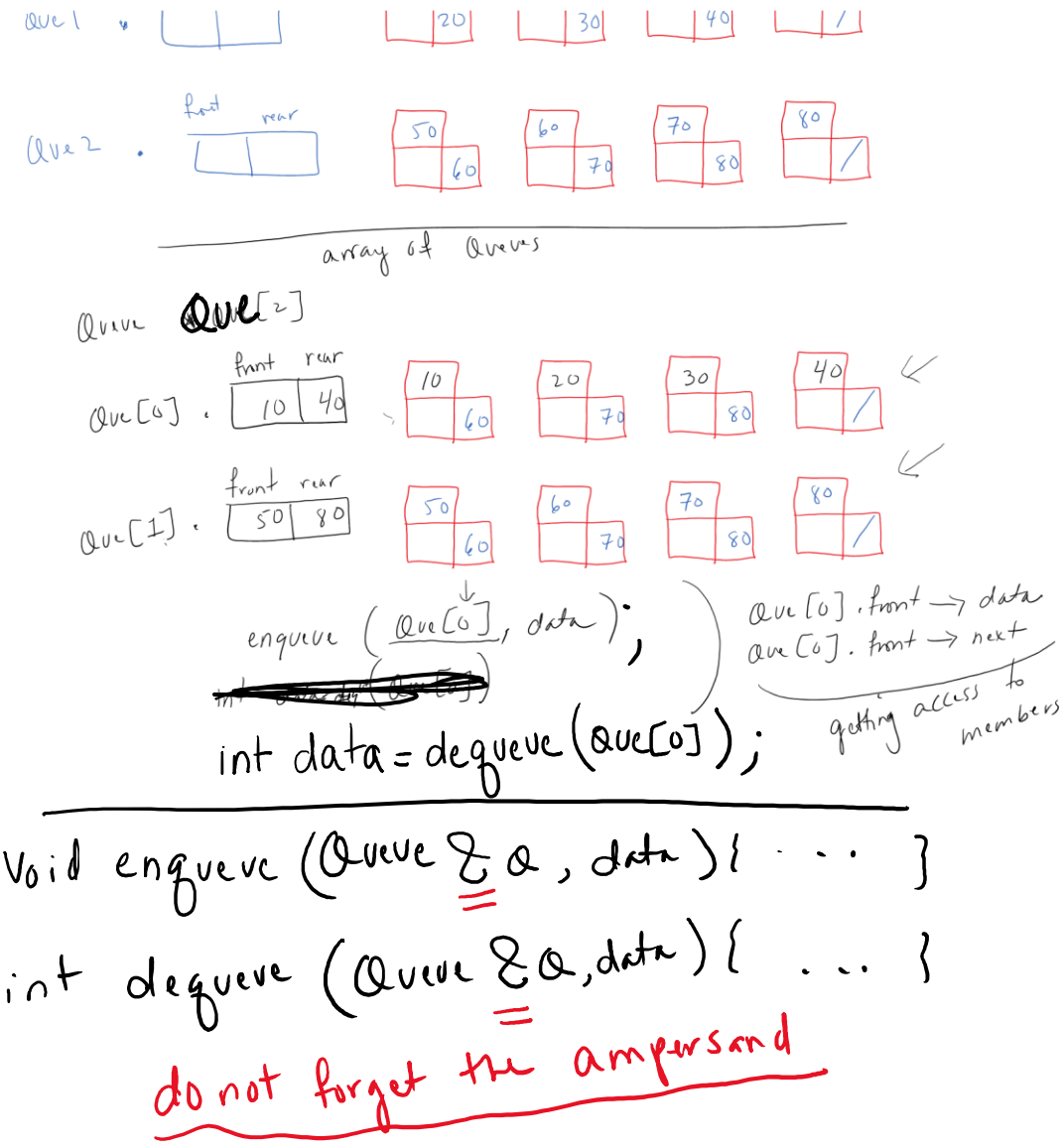
two Queues, One Structure

```
struct node {
    int data
    node * next
}
node * front1 , * rear1
node * front2 , * rear2
```



two Queues, One structure inside another structure

```
Struct node {
    int data
    node * next
}
struct Queue {
    node * front
    node * rear
}
struct Que1, Que2;
```


que1
20
30
40
que2
front
rear
50
60
60
70
70
80
80
array of Queues
Queue Que[2]
Que[0]
front rear
10 40
10
60
20
70
30
80
40
Que[1]
front rear
50 80
50
60
60
70
70
80
80
enqueue (Que[0], data);
Que[0].front → data
Que[0].front → next
getting access to members
int data = dequeue (Que[0]);
Void enqueue (Queue &q, data) { ... }
int dequeue (Queue &q, data) { ... }
do not forget the ampersand