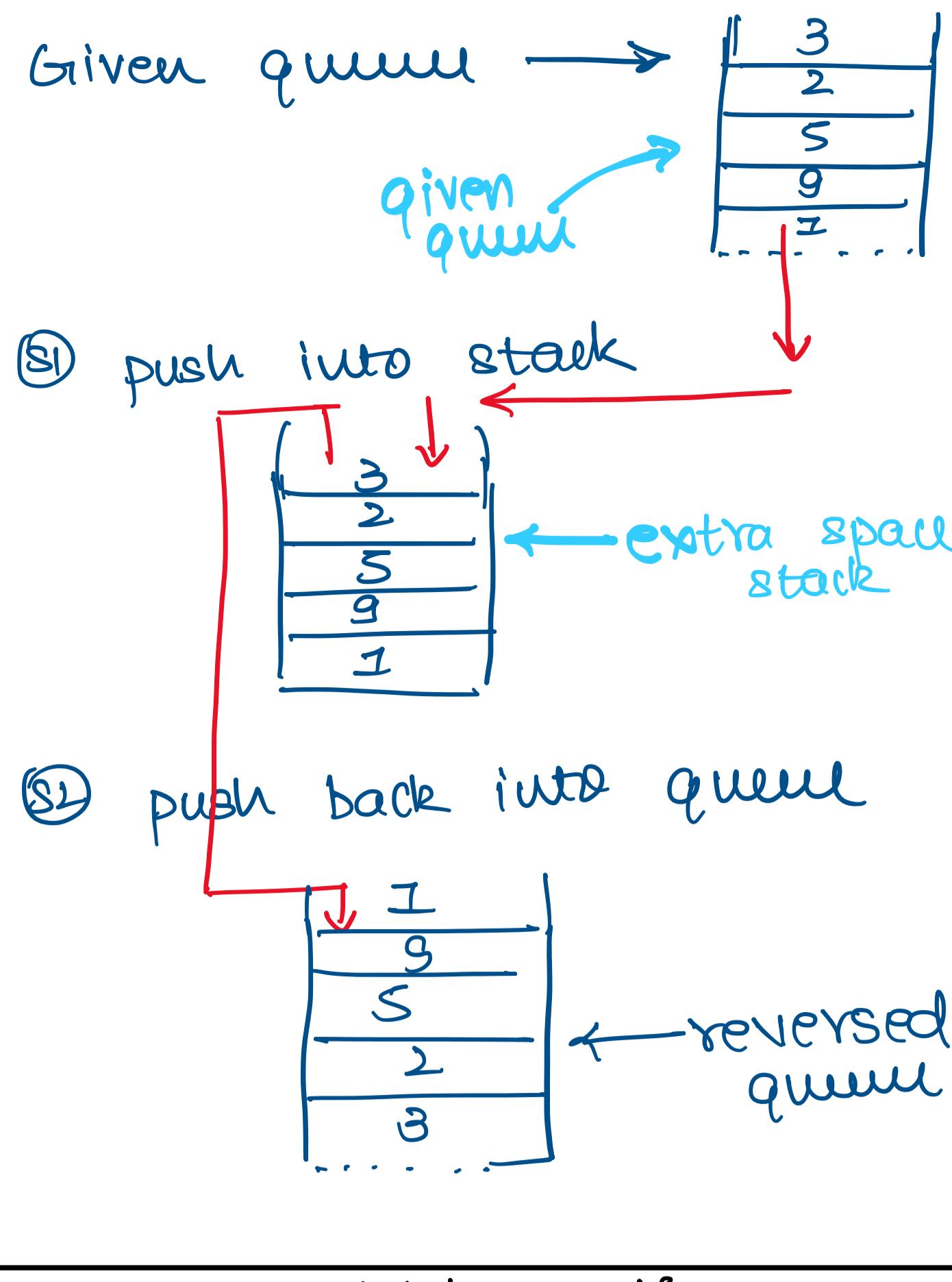


(Q) Reverse a Queue

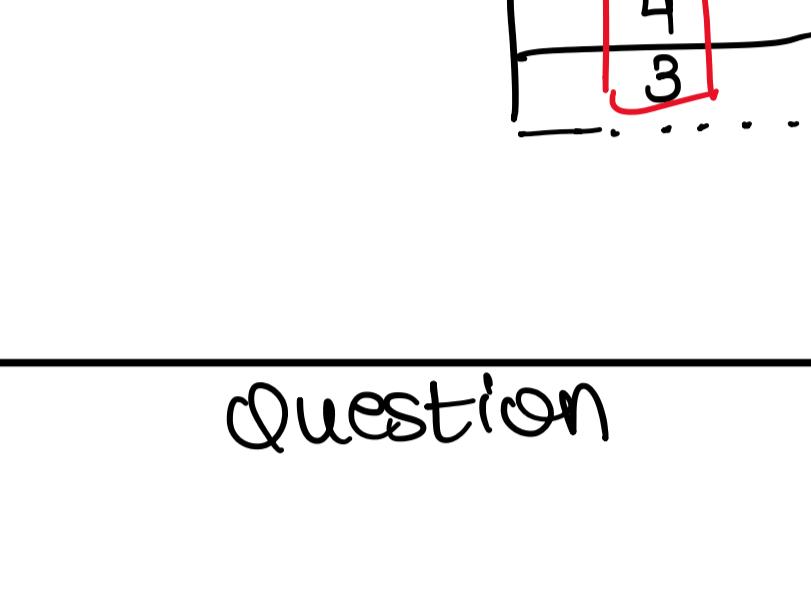
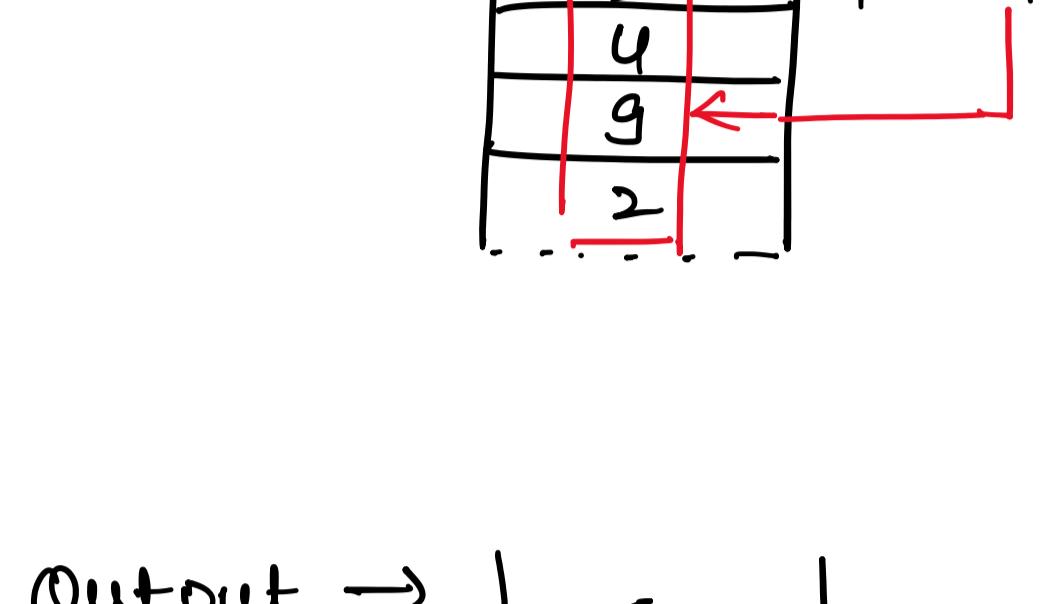


```
// Function to reverse the queue
void reverseQueue(queue<int>& Queue)
{
    stack<int> Stack;
    while (!Queue.empty()) {
        Stack.push(Queue.front());
        Queue.pop();
    }
    while (!Stack.empty()) {
        Queue.push(Stack.top());
        Stack.pop();
    }
}
```

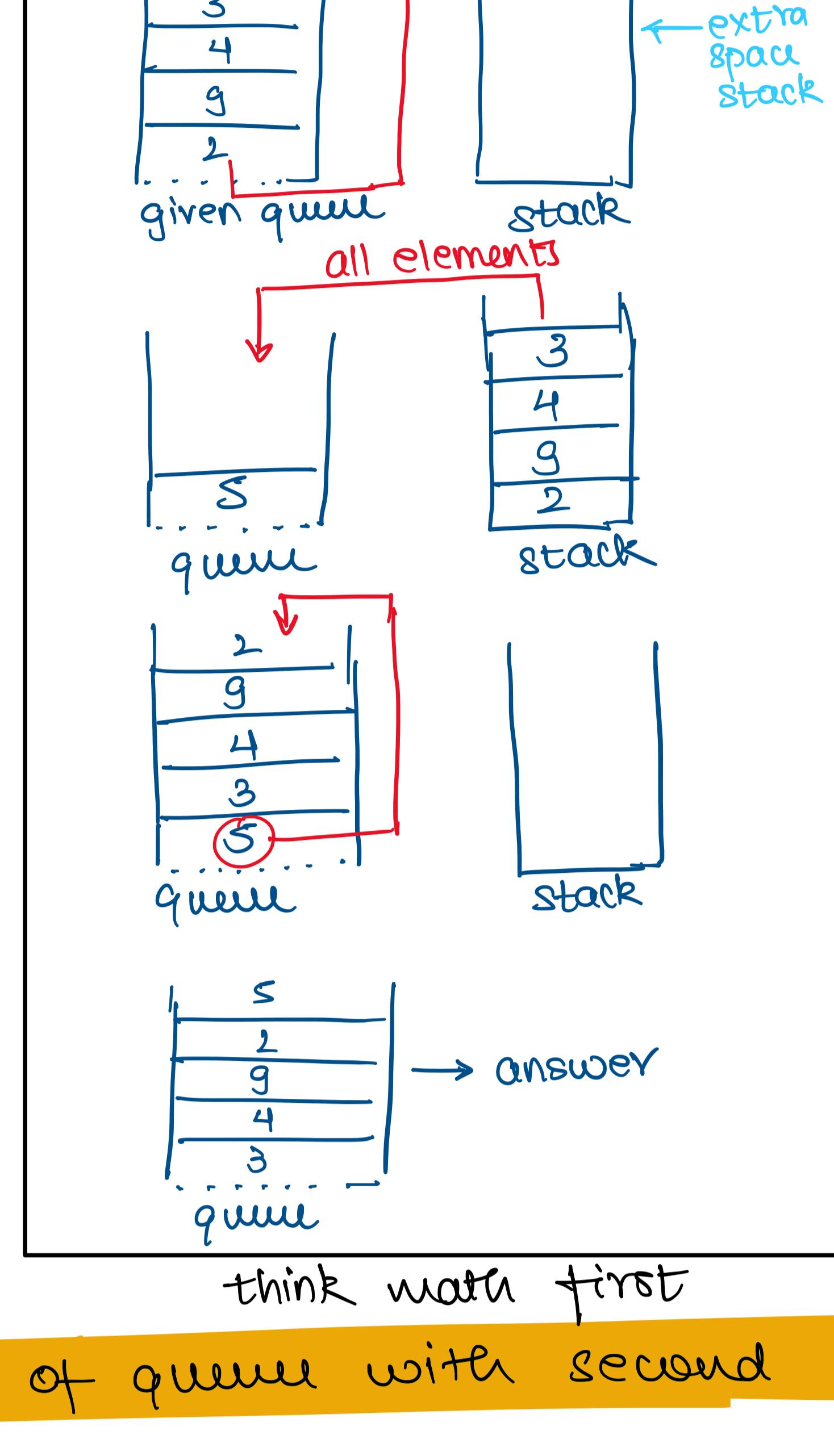
think math

then code

(Q) Reverse the first k elements of Queue



Question



```
void reverseQueueFirstKElements(int k, queue<int>& Queue)
{
    if (Queue.empty() == true || k > Queue.size())
        return;
    if (k <= 0)
        return;

    stack<int> Stack;

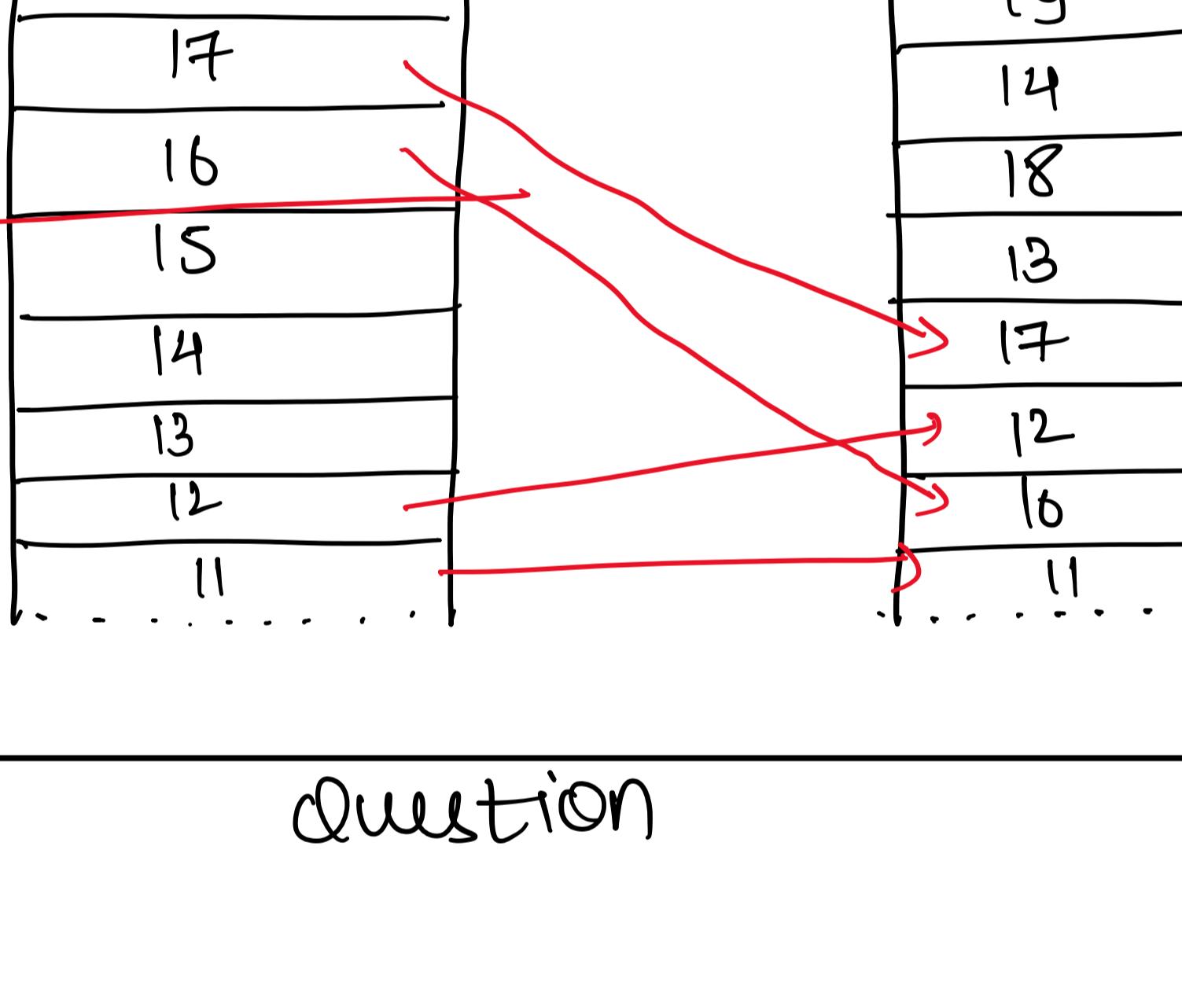
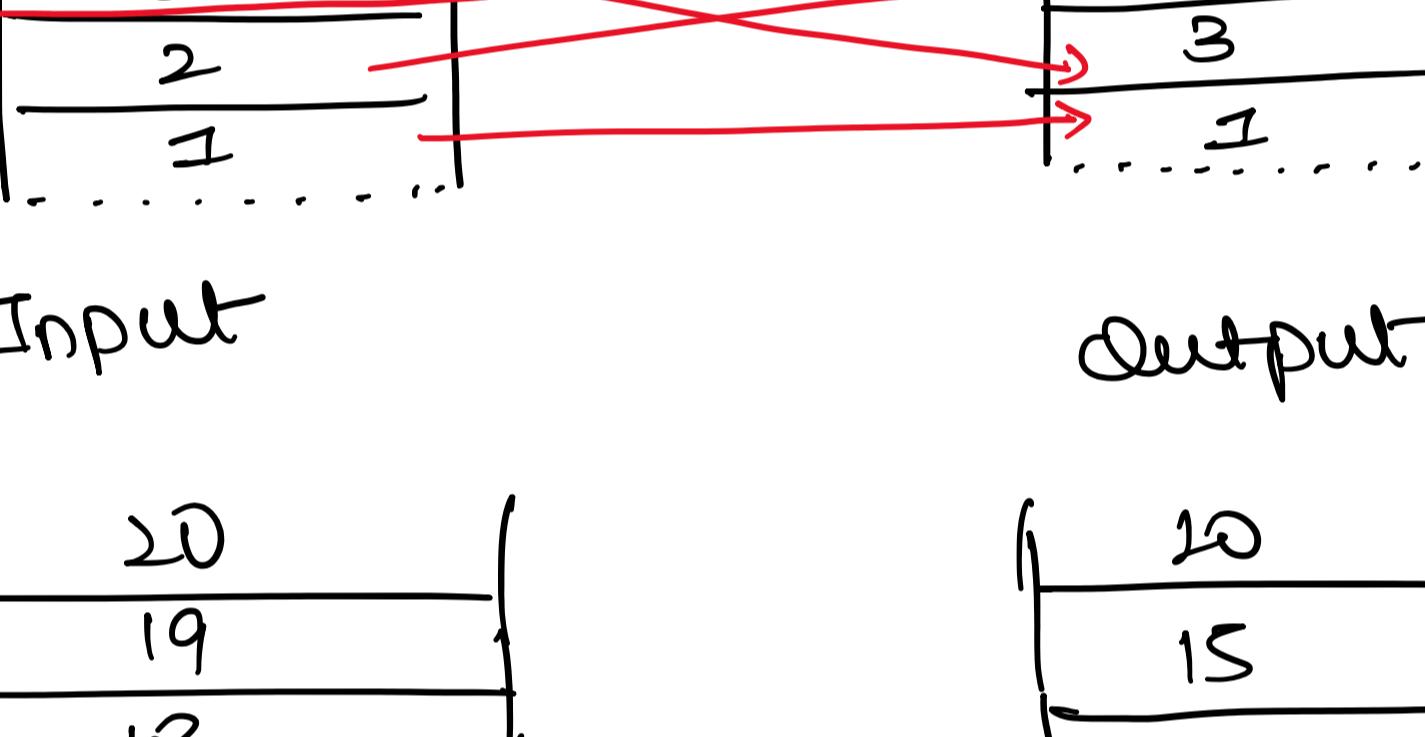
    /* Push the first K elements into a Stack*/
    for (int i = 0; i < k; i++)
    {
        Stack.push(Queue.front());
        Queue.pop();
    }

    /* Enqueue the contents of stack at the back of the queue*/
    while (!Stack.empty())
    {
        Queue.push(Stack.top());
        Stack.pop();
    }

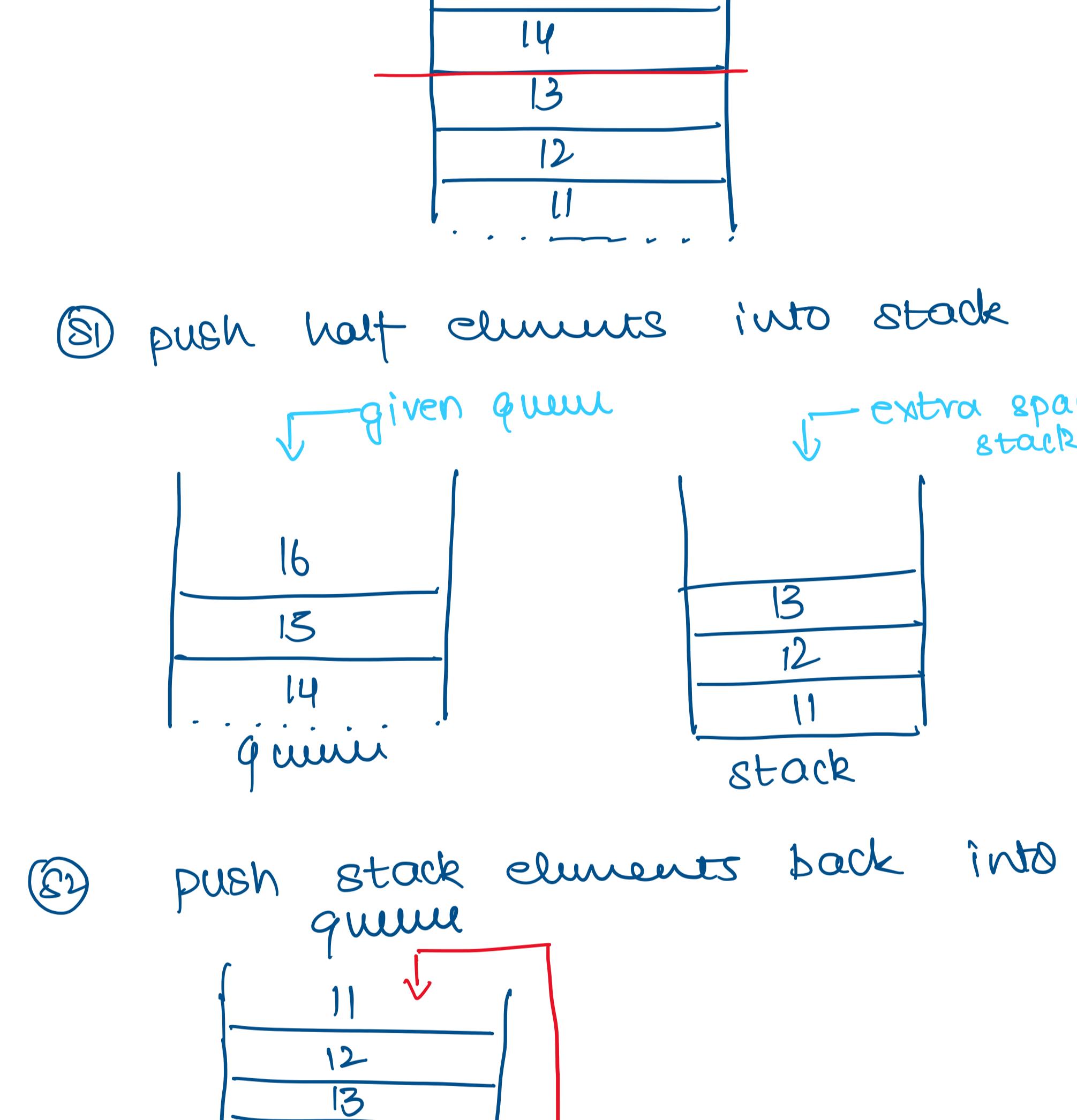
    /* Remove the remaining elements and enqueue them at the end of the Queue*/
    for (int i = 0; i < Queue.size() - k; i++)
    {
        Queue.push(Queue.front());
        Queue.pop();
    }
}
```

then code

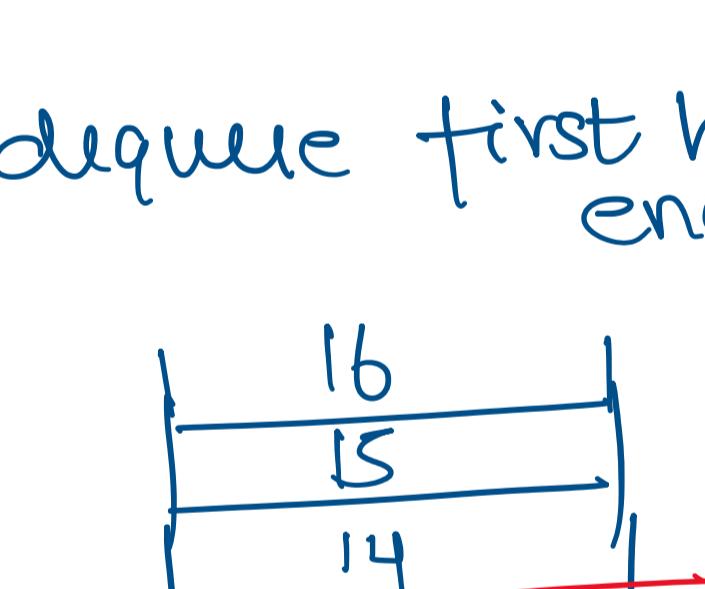
(Q) Interleave the first half of queue with second half



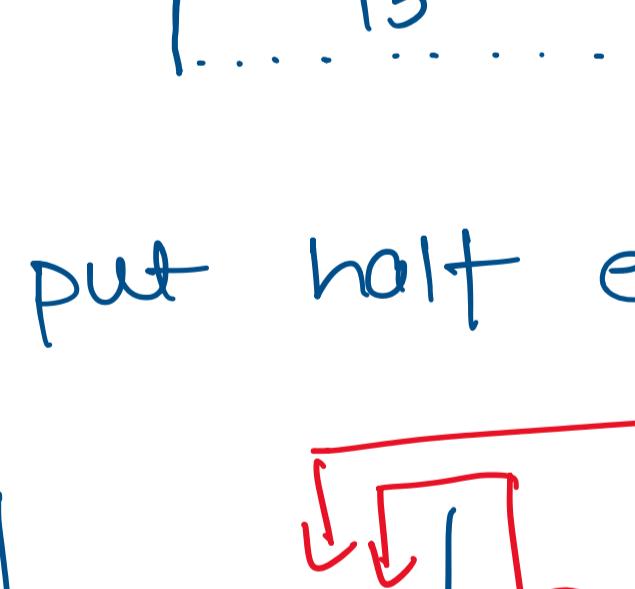
Question



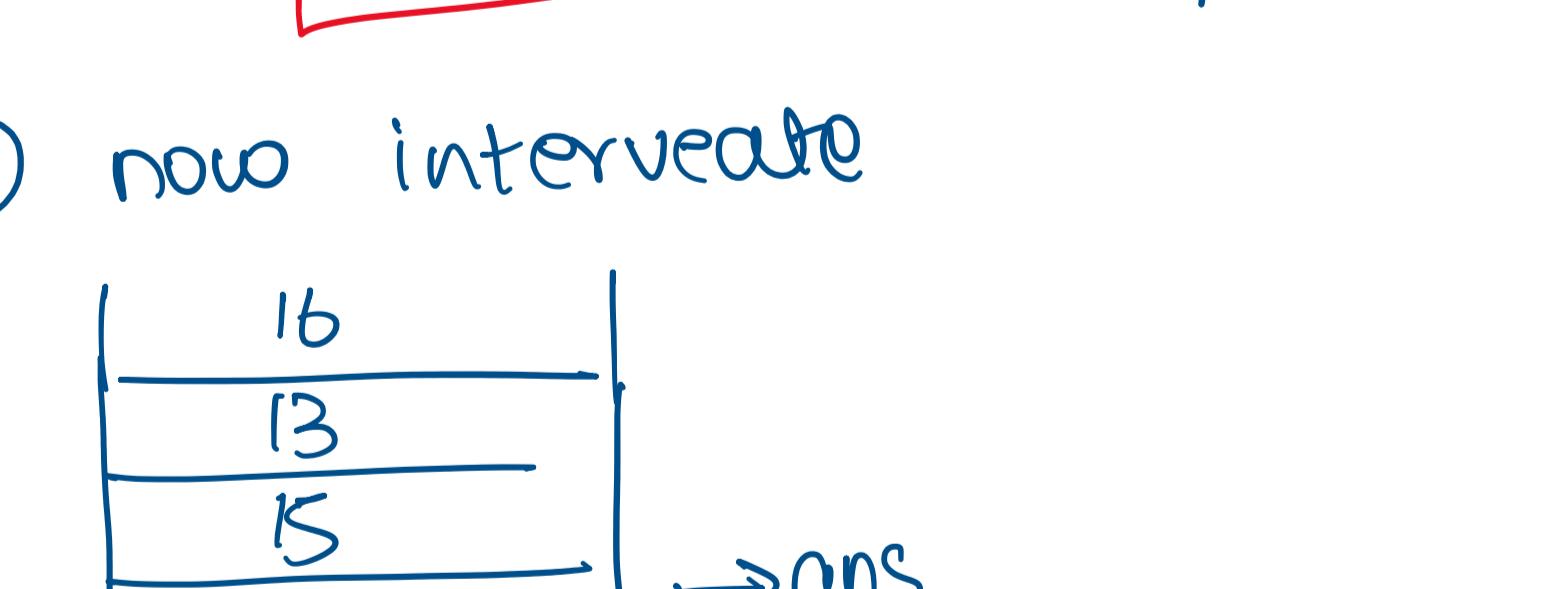
④ push stack elements back into queue



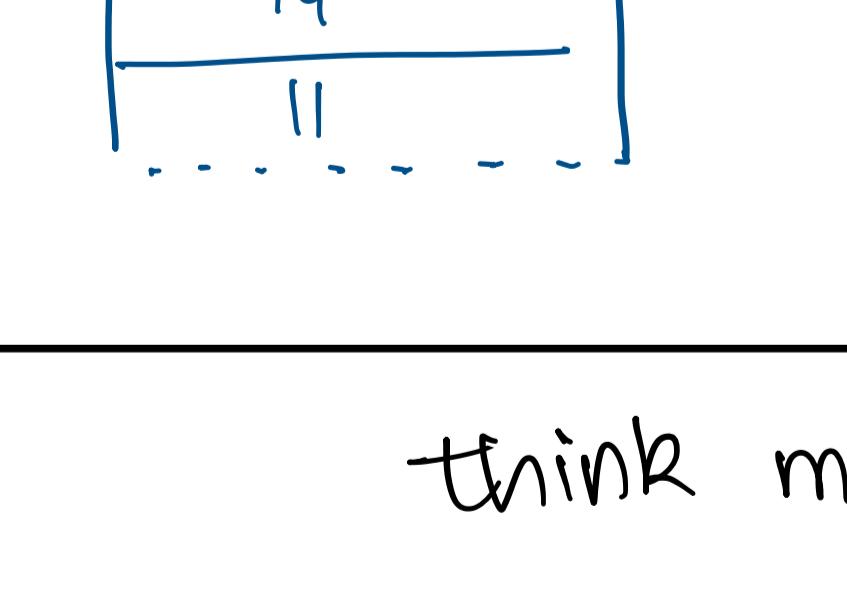
⑤ dequeue first half of queue and enqueue them back



⑥ put half elements back into stack



⑦ now interleave



think math first

```
void interLeaveQueue(queue<int>& q)
{
    // To check the even number of elements
    if (q.size() % 2 != 0)
        cout << "Input even number of integers." << endl;

    // Initialize an empty stack of int type
    stack<int> s;
    int halfSize = q.size() / 2;

    // Push first half elements into the stack
    for (int i = 0; i < halfSize; i++)
    {
        s.push(q.front());
        q.pop();
    }

    // enqueue back the stack elements
    while (!s.empty())
    {
        q.push(s.top());
        s.pop();
    }

    // dequeue the first half elements of queue and enqueue them back
    for (int i = 0; i < halfSize; i++)
    {
        q.push(q.front());
        q.pop();
    }

    // Again push the first half elements into the stack
    for (int i = 0; i < halfSize; i++)
    {
        s.push(q.front());
        q.pop();
    }

    // interleave the elements of queue and stack
    while (!q.empty())
    {
        q.push(s.top());
        s.pop();
        q.push(q.front());
        q.pop();
    }
}
```

then code