```
template<class T> class BoundedQueue {
public:
virtual ~BoundedQueue() { }
virtual bool isempty() = 0;
virtual bool isfull() = 0;
virtual void put(T& t) = 0; // add t to queue
virtual T get() = 0; // remove element from queue
};

template<class T>
class ArrayQueue: public BoundedQueue<T> {
ArrayQueue(int s) : BoundedQueue() {
        queue <T> myqueue;
        int n ;
        cout << “Input n – number of element in your array: “;
        cin >> n;
        BoundedQueue a[n];
        For (int I = 0 ; I < n ; i++
                a[i] = getArray(i);
        If (n < s)
                for (int i = 0 ; i < n; i++) myqueue.push_back(a[i]);
        BoundedQueue F,T;
        F = myqueue.front();
        T = myqueue,back();
}
};
```