```cpp
#include <queue>
#include <iostream>

using namespace std;

class MyStack {
public:
    MyStack() {

    }

    // Push element x to the top of the stack.
    void push(int x) {
        // Move all elements from mainQueue to tempQueue
        while (!mainQueue.empty()) {
            tempQueue.push(mainQueue.front());
            mainQueue.pop();
        }

        // Add the new element to mainQueue
        mainQueue.push(x);

        // Move elements back from tempQueue to mainQueue
        while (!tempQueue.empty()) {
            mainQueue.push(tempQueue.front());
            tempQueue.pop();
        }
    }

    // Removes the element on the top of the stack and returns it.
    int pop() {
        int topElement = mainQueue.front();
        mainQueue.pop();
        return topElement;
    }

    // Get the top element.
    int top() {
        return mainQueue.front();
    }

    // Returns whether the stack is empty.
    bool empty() {
        return mainQueue.empty();
    }

private:
    std::queue<int> mainQueue;
    std::queue<int> tempQueue;
```

```cpp
};

int main() {
    MyStack myStack;
    myStack.push(1);
    myStack.push(2);

    // Test pop
    int poppedValue = myStack.pop();
    cout << "Popped Value: " << poppedValue << endl; // Should print "Popped Value: 2"

    // Test top
    int topValue = myStack.top();
    cout << "Top Value: " << topValue << endl; // Should print "Top Value: 1"

    // Test empty
    bool isEmpty = myStack.empty();
    cout << "Is Stack Empty: " << (isEmpty ? "true" : "false") << endl; // Should print "Is Stack Empty: false"

    return 0;
}
```