HW1-哥尼斯堡七桥问题（查找欧拉回路）

实验结果：

```
请输入需检测回路大小
m = 4
n = 4
请输入需检测回路
0 1 2 2
1 0 1 1
2 1 0 0
2 1 0 0
无欧拉回路，因为有4个有奇数度的顶点
```

主要代码：

```cpp
int main(int argc, const char* argv) {
    //作业1：哥尼斯堡七桥问题(判断欧拉回路)
    HW1_SevenBridges hw1;
    hw1.CheckMat();
}
```

```cpp
class HW1_SevenBridges {
public:
    vector<vector<int>> mat;

    HW1_SevenBridges();
    ~HW1_SevenBridges();

public:
    void CheckMat();

private:
    void getMat();
    void EulerCircuit();
};
```

```cpp
void HW1_SevenBridges::CheckMat() {
    getMat();
    EulerCircuit();
}

void HW1_SevenBridges::getMat() {
    cout << "请输入需检测回路大小" << endl;

    int m, n;
    cout << "m = "; cin >> m;
    cout << "n = "; cin >> n;

    cout << "请输入需检测回路" << endl;

    for (int i = 0; i < m; i++){
        vector<int> temp;
        int tempNum;
        for (int j = 0; j < n; j++){
            cin >> tempNum;
            temp.push_back(tempNum);
        }
        mat.push_back(temp);
    }
}

//欧拉回路：能一笔画的封闭路径
void HW1_SevenBridges::EulerCircuit() {
    //记录有奇数边通过的顶点个数
    int cnt = 0;

    //计算有奇数边通过的顶点个数
    for (int i = 0; i < mat.size(); i++) {
        int degree = 0;
        for (int j = 0; j < mat[0].size(); j++) {
            degree += mat[i][j];
        }
        if (degree % 2 != 0) cnt++;
    }

    //判断欧拉回路条件：
    //  如果通奇数桥的地方多于两个，则不存在欧拉回路；
    //  如果没有一个地方通奇数桥，则无论从哪里出发，都能找到欧拉回路
    if (cnt <= 2) cout << "有欧拉回路" << endl;
    else          cout << "无欧拉回路，因为有" << cnt << "个有奇数度的顶点" << endl;

}
```