

**2021年春季学期  
计算学部《软件构造》课程**

**Lab 1实验报告**

|  |  |
| --- | --- |
| 姓名 | 王博 |
| 学号 | L190201901 |
| 班号 |  |
| 电子邮件 | wangbosheng@yandex.ru |
| 手机号码 |  |

**目录**

Catalog

**[1 实验目标概述 1](#_Toc5455)**

**[2 实验环境配置 1](#_Toc16586)**

**[1. Eclipse Java 1](#_Toc27586)**

**[2. Git 1](#_Toc2001)**

**[3 实验过程 1](#_Toc11561)**

[3.1 Magic Squares 1](#_Toc8007)

[3.2 Turtle Graphics 4](#_Toc30320)

[3.3 Social Network 8](#_Toc11419)

**[4 实验进度记录 11](#_Toc19096)**

**[5 实验过程中遇到的困难与解决途径 11](#_Toc5335)**

**[6 实验过程中收获的经验、教训、感想 12](#_Toc22629)**

[6.1 实验过程中收获的经验和教训 12](#_Toc14859)

[6.2 针对以下方面的感受 12](#_Toc5789)

# 实验目标概述

学习和了解：

1. 基本的 Java OO 编程
2. 基于 Eclipse IDE 进行 Java 编程
3. 基于 JUnit 的测试
4. 基于 Git 的代码配置管理

# 实验环境配置

1. Eclipse Java

先在官网下载java开发工具包JDK8 for windows，然后在官网下载Eclipse 都是一步步安装提示安装。其中安装Eclipse的时候出现提示后来发现是不能放在中文目录下面，然后又提示安装时间超出正常安装失败，可能是因为选择的那个镜像本来就很慢，就反复多试几次就安装成功了。

1. Git

根据GitHub的网站提示一步步安装就好。

GitHub Lab1仓库的URL地址：

https://github.com/ComputerScienceHIT/HIT-Lab1-L190201901

# 实验过程

## Magic Squares

这次任务的目的是要求设计一个程序：

任务一：能够从五个已经提供的txt文档中读入数据，来判断输入的数据能否构成一个MagicSquare，若是则返回true，若不是则返回false且说明原因。MagicSquare是一个正方形图形，他的每一行、每一列和对角线的数字之和都相等。

任务二：对给出的generateMagicSquare函数进行扩充，使之能够产生一个MagicSquare并且将其输入到文本6.txt中，然后对其判断是否为MagicSquare。并且满足若输入的为奇数能产生MagicSquare，而输入偶数或者负数会返回false。

### isLegalMagicSquare()

思路：函数能够读入文件中的数字来储存在矩阵中，所以首先需要判断读入的数据能否构成一个矩阵，根据实验手册，需要判断的三张错误分别为行列数不相等、并非矩阵，矩阵中某些数字不是正整数，不是以\t作为分隔符。对三种错误进行判断并输出返回false。当没有这些错误时，分别计算矩阵每一行、每一列、对角线的和是否相等来判断能否构成MagicSquare，并返回true。

过程：

对于三种错误：

行列数不相等、并非矩阵：我首先定义一个变量在读入每一行时进行加一的操作，来计算总行数，然后对每一列按照\t进行分割装在矩阵中，判断每一列是否与行数相等，若有一列不相等，则返回false。

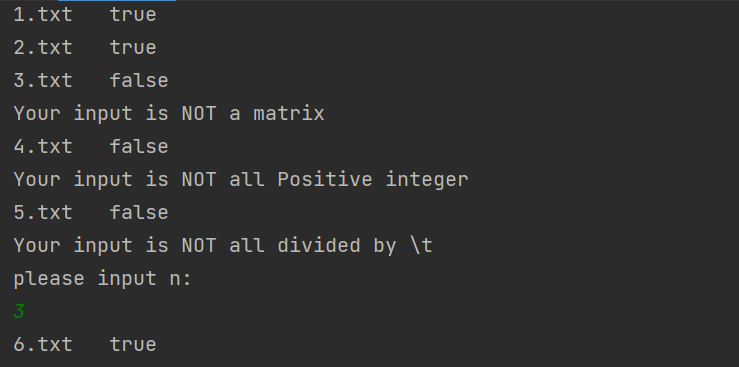
矩阵中某些数字不是正整数：与1中一样，首先按照\t将分割后的字符串装入数组中，然后判断这些字符串是否包含”.”或者”-”，若包含，则返回false。这样能够除掉包含小数和负数的情况。

不是以\t作为分隔符：若不是以\t作为分隔符，则会在Integer.valueOfs时抛出异常信息，并返回false。

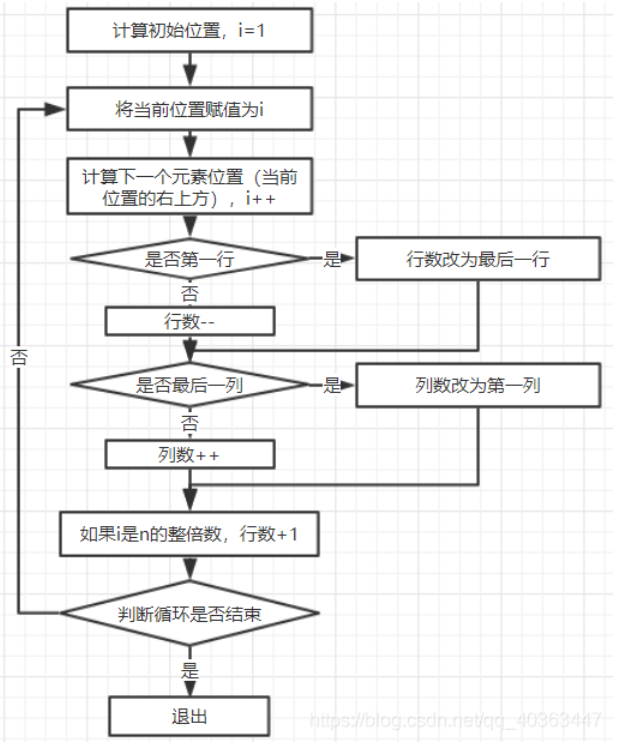
对于判断是否是MagicSquare定义如下：

分别计算每一行、每一列、对角线的和储存在相应数组中，首先判断三个数组的首元素是否相等，不相等返回false，然后在每一个数组内判断是否相等。最后返回true

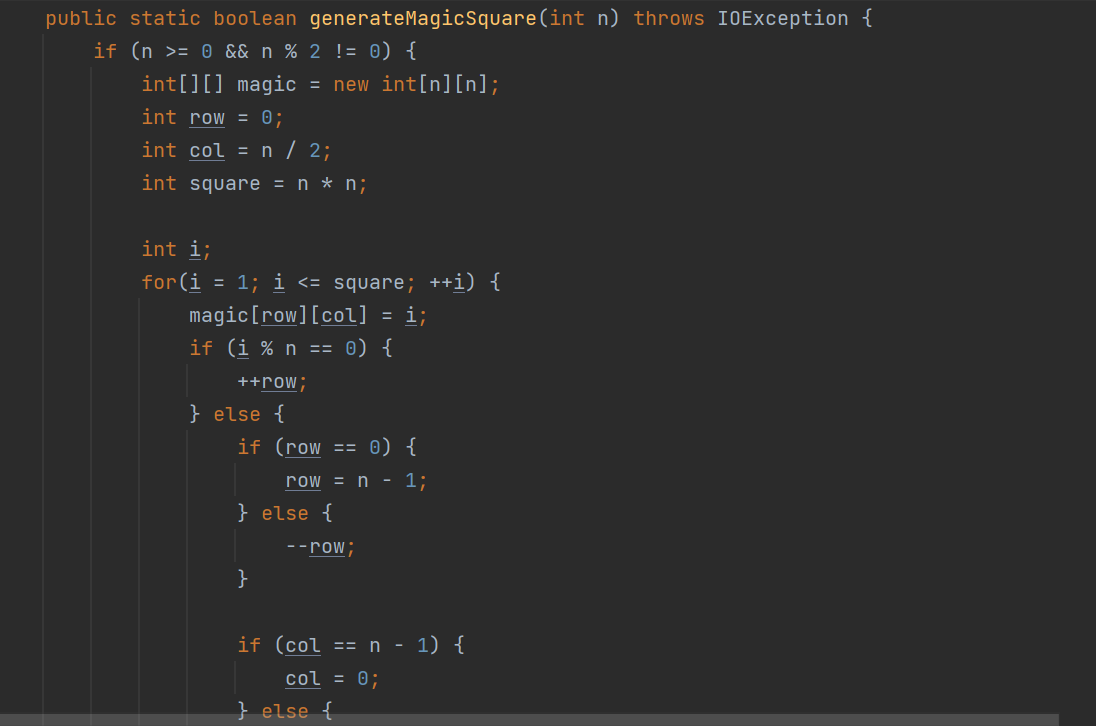
结果：

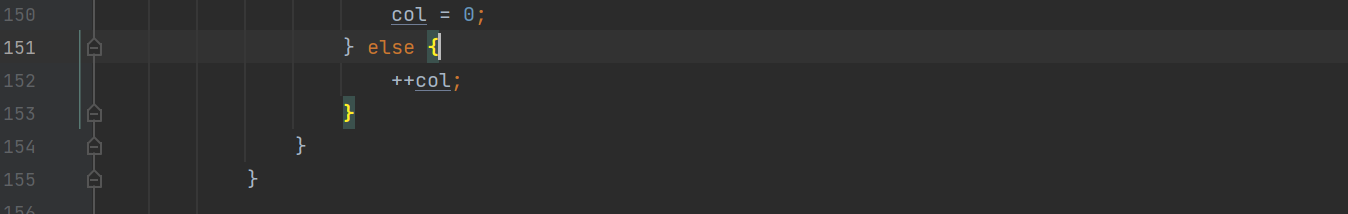


### generateMagicSquare()

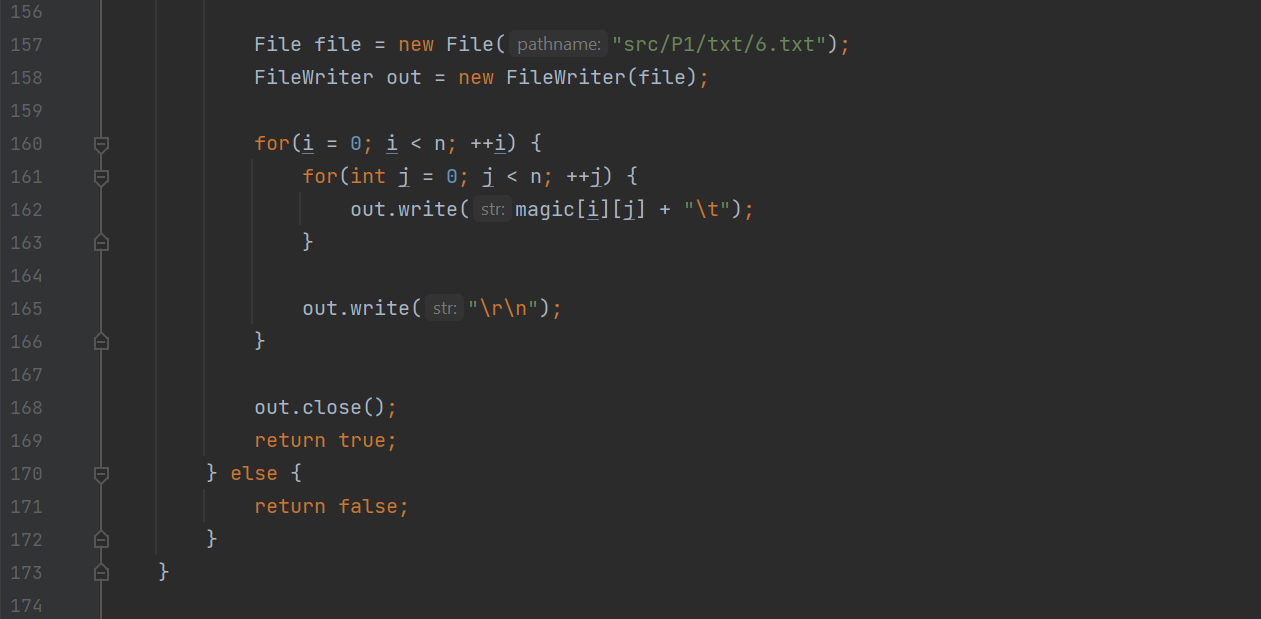


思路：首先计算初始位置（0，n/2），赋值为1，之后每次取当前位置的右上角的位置，设置的值每次加1，如果当前行是第一行，则下一行为最后一行，如果当前列是最右边的列，则下一次取左边第一列，如此重复平方次，就对整个矩阵赋值了，而且满足了每行每列以及对角线之和都是相同数字的效果。





完成矩阵的设置后输出矩阵成一个txt文件，代码如下：



## Turtle Graphics

任务就像是模拟python里面的turtle画图库的最最基本的功能，而且其中主要的功能函数和GUI界面都已经写好了，我们只需要完成TurtleSoup.java的编写：根据所给的函数绘制一些图案，按照要求一步步实现几个方法，然后重要的是学习junit测试的基础知识。

### Problem 1: Clone and import

Get clone 老师项目的网址 获取到代码，

然后复制到本地的lab1仓库，在Eclipse下完成编写任务后，命令行git here,（因为老师的项目里面只是代码而不是完整的项目，所有不能直接像MIT任务一样import git项目）

### Problem 3: Turtle graphics and drawSquare

forward(units)，直行，turn(angle)顺时针转弯，通过调用这两个方法画出正方形。很简单：

1. turtle.forward(sideLength);
2. turtle.turn(90);
3. turtle.forward(sideLength);
4. turtle.turn(90);
5. turtle.forward(sideLength);
6. turtle.turn(90);
7. turtle.forward(sideLength);
8. turtle.turn(90);

### Problem 5: Drawing polygons

首先是要完成calculateRegularPolygonAngle 这个方法，就是输入正多边形的边数求角度。值得注意的是这里需要不呢整除保留两位小数，能整除就是保留一位.0

用到的公式就是 每个内角角度=180\*（边数-2）/边数，保留精确度的方法是：(double) Math.round(angle \* 100) / 100.0

我这里遇到了一个问题就是用了angle = 180 \* (sides - 2) / sides;但是最后结果总是不对，后来发现这里会自动变成整数运算，所以需要改为angle = 180.0 \* (sides - 2) / sides;

calculatePolygonSidesFromAngle就是根据角度求边数。公式是：边数=360/（180-angle）这里还需要舍入到最近的整数，使用的是math.round()

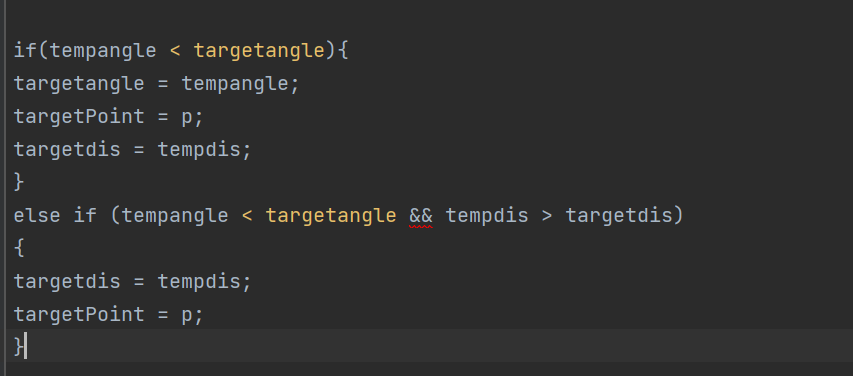
最后就是绘制这个多边形了，由于缺乏空间想象力，弄了好久才发现turtle每次转弯的话，需要转（180-angle）而不是直接angle

### Problem 6: Calculating Bearings

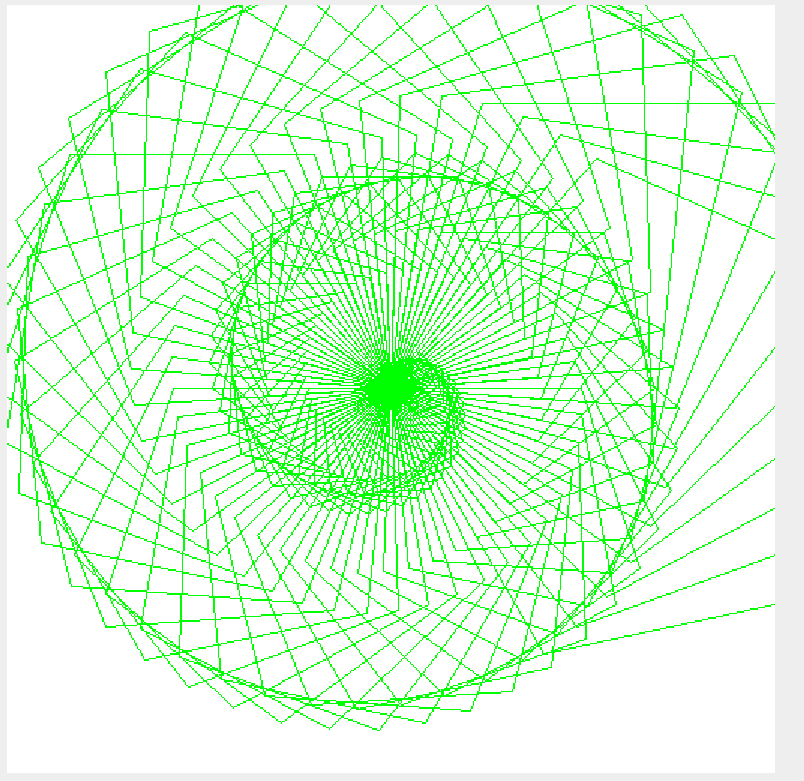
通过几何方法，利用反正切函数，可计算出结果。

### Problem 7: Convex Hulls

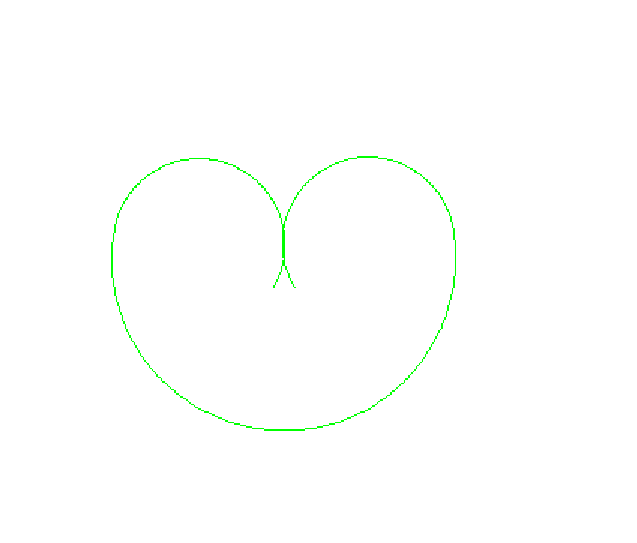
运用Gift wrapping algorithm算法，每次选择转向角最小的点（同时距离最长的点），加入集合，求转向角函数直接使用的之前写好的calculateBearingToPoint函数，



### Problem 8: Personal art



或



其中使用PenColor color = PenColor.***GREEN***;

turtle.color(你的颜色的); 来变换颜色

### Submitting

依此使用get add .（把工作时的**所有变化提交**到暂存区），

git commit -m 'L190201901' (提交暂存区的文件),

git pull origin master(命令用于取回远程主机master分支的更新与本地的当前分支合并)，

git pull origin master（将当前分支推送到origin主机的对应分支。）

如何通过Git提交当前版本到GitHub上你的Lab1仓库。

## Social Network

本次任务要求实现Person和FriendshipGraph两个类，用FriendshipGraph来构建Person之间的关系来模拟社交网络，能够计算出每两个Person之间的最短路径。

### 设计/实现FriendshipGraph类

思路：在FriendshipGraph类中我们要完成的功能为：增加一个人的对象addVertex（若是增加的人为已存在的则输出“此名已存在，重复”并结束程序），增加一条有向边addEdge，得到两个人之间的最短距离getDistance。对此三个功能的实现，主要是功能三的实现相对复杂，我们决定采取先广的方式求最短路径。

过程：

1.定义一个Person集合来储存所有人的对象，定义一个String集合来储存所有已经存在的名字：

2.定义构造方法对两个集合进行初始化：

3.实现功能一：addVertex。实现此功能需要注意用来增加的Person对象是否已经存在，所以需要在开始判断allname集合中是否已经有了待增加的Person对象的名字，若已经存在，则输出“此名已存在，重复”并结束程序，若没有，则将待增加的新的Person对象加入allpeople集合中，并且将对象的名字加入allname集合中：

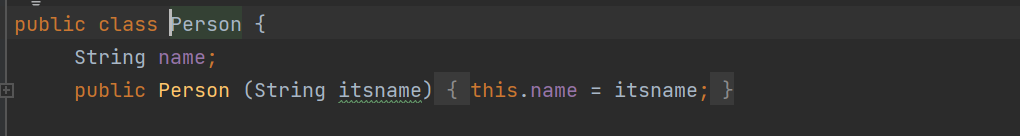
4.实现功能二：addEdge。实现此功能只需要调用Person类的addnewfriend方法即可：

5.实现功能三：getDistance。实现此功能我们需要了解先广方式求最短路径的方法。首先我们需要知道若两个Person对象为同一个，则返回0。然后定义一个Map集合theway和一个Person队列myqueue，队列myqueue用来储存广搜的遍历结果，theway的Map集合用来储存广搜的所有元素及他们与第一个元素的距离。具体实现方法为首先将第一个元素c1入队，并且把第一个元素c1和下标0入集合，当队列非空时，弹出队首元素top，并且得到top在集合theway中的下标distance，然后调用gethisfriend函数得到队首元素的所有朋友allfriend，只要allfriend中的元素与c2不同，就把这些元素全部入队，并且把这些元素及下标distance+1放入集合。只要队列非空，继续执行以上步骤，直到找到某个元素与c2相同并且返回这个元素在集合theway中的下标。如果直到队列为空还没找到c2，则返回-1

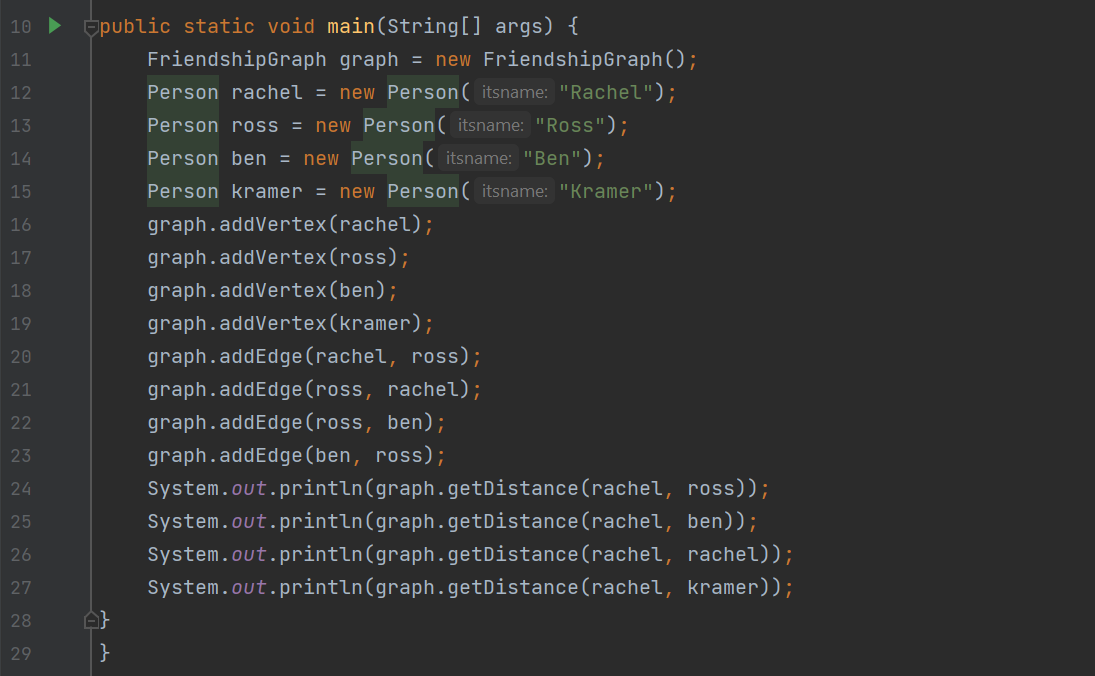


### 设计/实现Person类

Person类里面只用来储存这个人的名字和所以只有一个String和一个构造函数



### 设计/实现客户端代码main()



Main函数放在Person类里面，是用题目给出的例子来测试结果。

### 设计/实现测试用例

题目给出的测试用例：

Main函数输出结果是 0 2 0 -1 是符合预期的

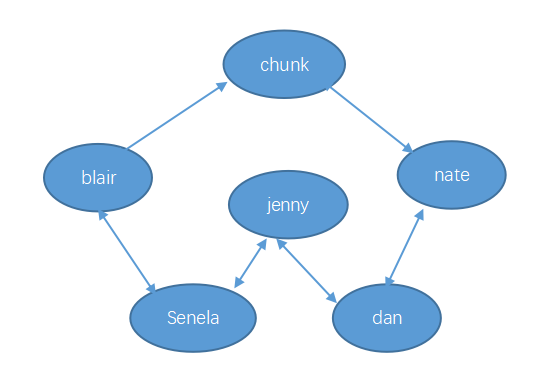
其他测试：

1.如果 如果将上述代码的第 10 行注释掉（意即 rachel 和 ross 之间只存在单向的 社交关系 ross->rachel）那么输入的结果是 -1 -1 0 -1也满足预期，因为此时ross不认识rachel,那么ross认识ben，rachel也不会认识ben,此时关系不相互不能传递。

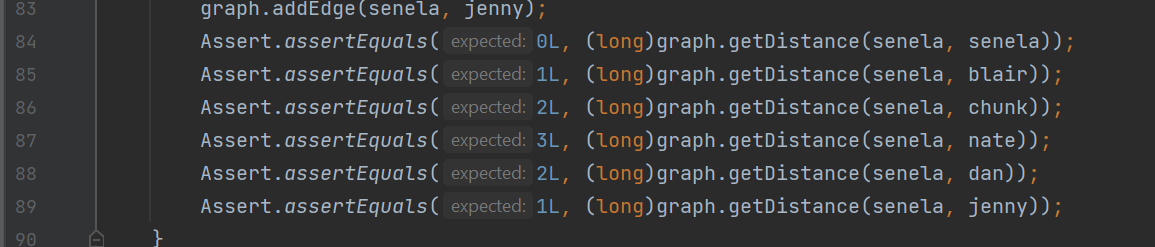
2.如果将第 3 行引号中的“Ross”替换为“Rachel”，那么程序会报错，因为题目默认每个人只有一个名字，而列表里面已经有rachel了。

3.自己写的测试用例：

（用图表示）



输出结果:



# 实验进度记录

请使用表格方式记录你的进度情况，以超过半小时的连续编程时间为一行。

每次结束编程时，请向该表格中增加一行。不要事后胡乱填写。

不要嫌烦，该表格可帮助你汇总你在每个任务上付出的时间和精力，发现自己不擅长的任务，后续有意识的弥补。

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 时间段 | 任务 | 实际完成情况 |
| 2021-05-20 | 17:30-19:30 | 编写问题1的isLegalMagicSquare函数并进行测试 | 按计划完成 |
| 2021-05-21 | 10:00-11:00 | 完成了第二个问题并上交github | 延期1小时完成 |
| 2021-05-21 | 11:00-13:00 | 完成第三个问题 | 遇到困难，未完成 |

# 实验过程中遇到的困难与解决途径

So hard, but so intersting!!!

没什么偶到困难！不是这样。一定有偶到困难但现在我知道了这个不是困难这就是基本的问题。从现在我终于懂了我就应该读多书吧！

# 实验过程中收获的经验、教训、感想

## 实验过程中收获的经验和教训

## 针对以下方面的感受

1. Java编程语言是否对你的口味？
2. 关于Eclipse IDE；
3. 关于Git和GitHub；
4. 关于CMU和MIT的作业；
5. 关于本实验的工作量、难度、deadline；
6. 关于初接触“软件构造”课程；