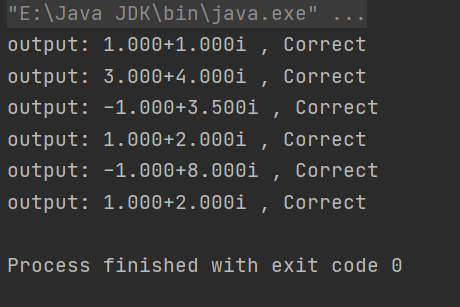
Report – HW2

段易非 2018010303

1. Complex



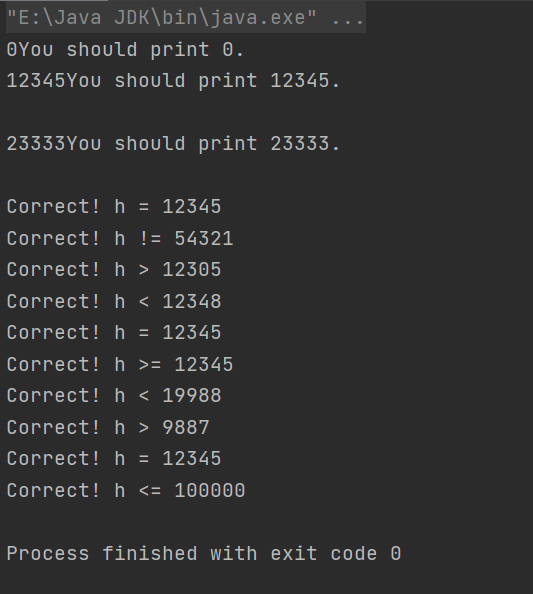
【代码实现】

用两个double类型变量来存储实部realPart和虚部imaginaryPart，构造函数直接将输入赋值给二者。对于四则运算的四个函数，根据复数运算规则分别对实部、虚部改变赋值。特别地，在乘法、除法中由于实部虚部的运算要用到彼此的值，因此不能直接在realPart、imaginaryPart上进行操作，而要声明新变量来保存初始的A\_real、A\_imaginary。

【问题解决】

三位小数的保留，使用了String.format(“%.3f”)函数。

1. HugeInteger



【代码实现】

由于大整数可达到40位，超过int的范围，因此考虑用整型数组来存放大整数的各位数码，在初始化时先把各位赋值为0再根据输入将低位重新赋值。为了最后输出打印结果时无多余位的0，因此单独声明一个变量来记录大整数有效的位数长度，在做各种操作时都要随时记录其变更信息。

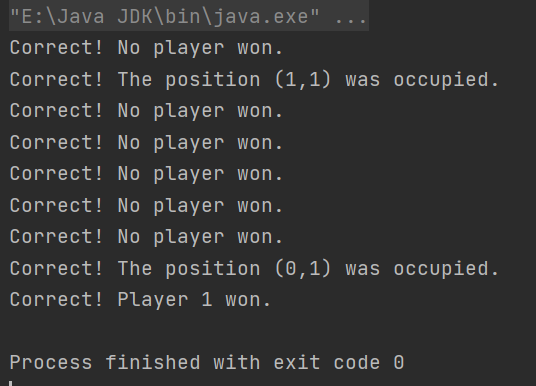
相加运算的add函数，从最低位开始逐位进行两数码加和，大于9则进位，由此完成相加。此后，还要查找首个不为0的数位，从而更新长度信息。

对于其他各种涉及大小比较的函数，统一使用一个compare函数实现。Compare函数的返回值满足：A>B则返回1，A<B则返回2，A=B则返回0。其设计思路是先比较A、B长度，有差异则可直接确定返回1/2；相同则从最高位开始利用for循环依次比较二者各位数码，一旦出现不同即可确定返回1/2，反之若一直比到最低位循环结束时仍未返回，则说明各位数码均一致，最终返回0。

【问题解决】

System.out.println()输出会换行，应使用System.out.print()将数组在一行内连续输出

1. TicTacToe



【代码实现】

声明3\*3二维数组board来保存棋盘信息，初始化为全0，表明均无棋子。

构造place函数来记录每步走棋的结果。首先，若该位置已被标记（不为0）则返回3，否则就在该位置标记上棋手（1/2）。接下来利用下面要提到的win函数判断棋手是否获胜，从而返回1/2，若仍未决出胜负则返回0。

构造win函数判断是否有人取胜，参数为player（1/2），返回值亦然。只需逐行、逐列、逐对角线判断是否均被该player标记占据，若有任一为是则可立即判定其获胜，直接返回。

【问题解决】

若使用循环语句初始化二维数组会报错，查阅资料后改采用:

new int[][]{ {0,0,0},{0,0,0},{0,0,0} };