

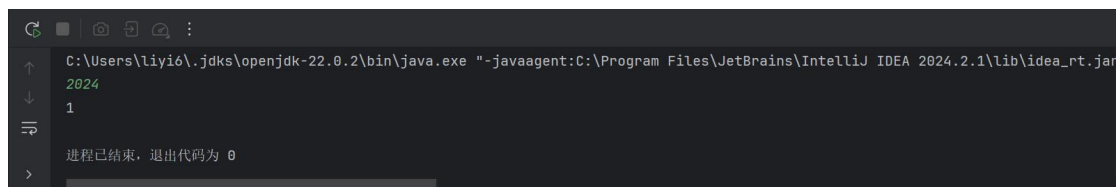
Task1

补全的代码如下：

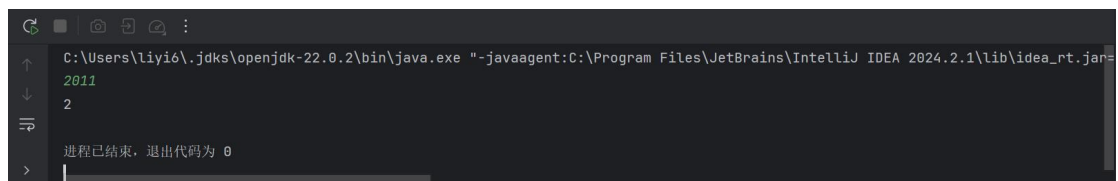
```
bool isLeapYear(int year) {  
    if (year % 400 == 0) {  
        return 1;  
    }  
    else if (year % 4 == 0 && year % 100 != 0) {  
        return 1;  
    }  
    else {  
        return 2;  
    }  
}
```

说明：

1. Java 中没有 bool 关键字，在运行程序时已改为 public static boolean isLeapYear(int year)
2. 第一、二个 if/else if 语句分别表示判断闰年的两个条件：
 - (1) ：年份能被 400 整除的是闰年
 - (2) ：年份能被 4 整除但不能被 100 整除的是闰年
3. else 语句输出不满足以上两个条件的不是闰年
4. 附：以 2024 年和 2011 年为例，完善后分别输出以下结果的截图



```
C:\Users\liy16\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.1\lib\idea_rt.jar=2024  
1  
进程已结束，退出代码为 0
```



```
C:\Users\liy16\.jdk\openjdk-22.0.2\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA 2024.2.1\lib\idea_rt.jar=2011  
2  
进程已结束，退出代码为 0
```

Task2

补全的函数为：

```
void print (int n) {  
    if (n % 2 == 0){  
        System.out.println("n 不为奇数");  
        return;  
    }  
    int middle = n / 2;  
    for (int a = 0; a < n; a++) {  
        for (int b = 0; b < n; b++) {  
            if (a <= middle) {  
                if (b == middle - a || b == middle + a) {  
                    System.out.print("*");  
                } else {  
                    System.out.print(" ");  
                }  
            } else {  
                if (b == a - middle || b == n - 1 - (a - middle)) {  
                    System.out.print("*");  
                } else {  
                    System.out.print(" ");  
                }  
            }  
        }  
        System.out.println();  
    }  
}
```

说明：

1. 最开始的一个 if 语句用来判断 n 是否为奇数，如果不是则返回“n 不为奇数”
2. 后面的一个嵌套 for 循环则先计算半高度（中心值 middle），通过遍历行数 a 和列数 b，a 在小于半高时，打印菱形上半（正金字塔型）通过判定列数 b 满足中值 middle ± 行数 a 来打印*，否则打印空格，这样随 a 的自增，空格数也递增，形成上半部分。下半部分同理，只不过空格数随 a 的递增而减少，形成下半部分（倒金字塔）。每行打印完后使用 println 换行。
3. 附：完善代码后通过外部输入高度的运行结果图（含 n 不为奇数时的报错）：

