实例 4：随机分配办公室

某学校新招聘了 8 名教师，已知该学校有 3 个空闲办公室且工位充足，现需要随机安排这 8 名教师的工位。

本实例要求编写程序，将 8 名教师随机分配到 3 个办公室中。

提示：随机选择办公室，可以使用 random.randint(0,2)实现，需使用 import random 导入 random 模块。

实例目标

- 掌握列表的嵌套使用

实例分析

本实例中的学校包含 3 个空闲的办公室，其中的每个办公室可以随意容纳教师。由于学校和办公室分别用于存储办公室与教师，且它们中数据的个数是可变的，因此可以用列表来表示学校和办公室，用嵌套列表表示学校与办公室的包含关系，此时表示第一个办公室的空列表的索引为 1，表示第二个办公室的空列表的索引为 2，表示第三个办公室的空列表的索引为 3。

随机分配办公室就是将每名老师逐个安排到任意的办公室中，这个过程可拆分为两步，第一步就是逐个取出教师姓名，相当于遍历列表元素的操作；第二步就是安排到任意的办公室，相当于使用 random 模块中 randint()方法生成随机 0-2 之间的整数，将产生的整数作为索引来随机获取嵌套列表的内层列表，之后在该列表中执行添加教师姓名的操作。

代码实现

本实例的具体实现代码如下所示。

```
import random
# 定义一个列表用来保存 3 个办公室
offices = [[], [], []]
# 定义一个列表用来存储 8 位老师的名字
names = ['张老师', '李老师', '赵老师', '高老师',
    '刘老师', '周老师', '王老师', '吴老师']
for name in names:
    # 将 8 位老师按照索引为 0、1、2 进行分组
    index = random.randint(0, 2)
    # print(index)
```

```
        # 将 8 位老师放在不同列表中
        offices[index].append(name)
    flag = 1
    for tempNames in offices:
        print('办公室%d 的人数为：%d' % (flag, len(tempNames)))
        flag += 1
        for name in tempNames:
            print("%s" % name, end=' ')
        print(" ")
```

以上代码首先定义了一个包含 3 个办公室的嵌套列表 offices，定义了另一个包含 8 位教师姓名的列表 names，然后将生成的 0~2 之间的随机数作为索引获取嵌套列表 offices 中的任一内层列表，将遍历 names 取出的元素添加到该内层列表中，直至遍历出最后一个元素为止，最后输出每个办公室的教师分配情况。

需要注意的是，为了美化输出的结果，这里定义了一个表示办公室编号的变量 flag，输出每个办公室的总人数与教师姓名。

代码测试

运行程序，程序运行的结果如下所示。

```
办公室 1 的人数为：3
张老师 刘老师 王老师
办公室 2 的人数为：1
赵老师
办公室 3 的人数为：4
李老师 高老师 周老师 吴老师
```

再次运行程序，程序运行的结果如下所示。

```
办公室 1 的人数为：3
高老师 周老师 吴老师
办公室 2 的人数为：3
张老师 李老师 刘老师
办公室 3 的人数为：2
赵老师 王老师
```