**姓名： 学号： 班级：**

**一、一个列表有四个元素，其中三个是a、b和c，第四个元素d是学号，组成互不相同4个数字和字母的排列组合。请补充横线处语句，粘贴运行结果。**

**count = 0**

**d = str(学号 % 10)**

**ls = ['a', 'b', 'c', d]**

**n = len(ls)**

**for i in range(n):**

**for k in range(n):**

**for m in range(n):**

**if ( ) and ( ) \**

**and (i!=m) and (j!=m) \**

**and (\_\_\_\_) and (\_\_\_\_):**

**print(ls[i]+ls[j]+ls[k]+ls[m], end=" ")**

**count += 1**

**print(count)**

**二、企业发放的奖金根据利润提成。**

**利润I低于或等于310万元时，奖金可提10%；**

**利润高于310万元，低于320万元时，低于310万元的部分按10%提成，高于310万元的部分，可提成7.5%；**

**320万到340万之间时，高于320万元的部分，可提成5%；**

**400万到410万之间时高于40万元的部分，可提成3%；**

**410万到420万之间时，高于60万元的部分，可提成1.5%，**

**高于420万元时，超过100万元的部分按1%提成，从键盘输入当月利润I，求应发放奖金总数？**

**请补充横线上的代码，要求输入的money是学号的后三位。粘贴运行结果。**

**money = float(\_\_\_\_\_\_("请输入当月利润，单位为万元："))**

**bonus = 0**

**li = list(range(100,40,-10))**

**li = [ ]**

**bi = [i\*0.01 for i in range(5,11,1)]**

**bi = [ ]**

**for i in range(6):**

**if money > li[i]:**

**cmoney = money - li[i]**

**bonus = bonus + cmoney \* bi[i]**

**money = li[i]**

**print("当月应发放奖金总数为%s万元" %\_\_\_\_\_)**

**三、求1到（200+学号）中所有奇数的和、以及所有偶数的和。补充横线处语句，粘贴运行结果。**

**sum\_odd = 0; sum\_even = 0**

**for i in range(1, 200+学号):**

**if**

**sum\_odd += i #奇数和**

**else:**

**sum\_even += i #偶数和**

**print("1~1\*\*中所有奇数的和:", sum\_odd)**

**print("1~1\*\*中所有偶数的和:", sum\_even)**

**四、判断输入的正整数（200+学号）是否为素数。补充横线处语句，粘贴运行结果。**

**import math**

**m = int(input("请输入一个整数："))**

**k = int(math.sqrt(m))**

**flag = True #先假设所输整数为素数**

**i = 2**

**while (\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_):**

**if (m % i == 0): flag = False #可整除，非素数，结束循环**

**else: i += 1**

**if (flag == True): print(m, "是素数！")**

**else: print(m, "是合数！")**

**五、猜数字,5次机会。补充横线处语句，粘贴运行结果。**

**import random**

**def guess\_number():**

**num = random.randint(0,40) # 生成一个从0-40的随机整数**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**for count in range(\_\_\_\_\_\_):**

**if a<=0 or a>=40:**

**print("专注考试, 请重新运行代码")**

**break**

**if a == num:**

**print('恭喜你猜对了，就是',a)**

**break**

**elif a > num:**

**print('第%d次猜数字，猜大了'%(count+1))**

**a = int(input('请重新输入一个数字:'))**

**elif a < num:**

**print('第%d次猜数字，猜小了'%(count+1))**

**a = int(input('请重新输入一个数字:'))**

**count += 1**

**else:**

**print("已经超过%d次了。"%count)**

**print("The turth number is %d"%num)**

**guess\_number()**

**六、要求完成横线代码，输入年份为学号后4位数字+1700，月份为学号后两位数字之和，日期是学号的后两位 。补充代码，运行程序，把输出结果截图放在文件最后。**

**#判断某年是否为闰年**

**def leapYear(y):**

**if y<1: y=1**

**if y % 400 == 0:**

**return 1**

**elif \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_:**

**return 1**

**else:**

**return 0**

**#计算y年m月的天数**

**def getLastDay(y,m):**

**if y<1:y=1**

**if m<1:m=1**

**if m>12:m=12**

**# 每个月的正常天数**

**# 月份 1 2 3 4 5 6 7 8 9 10 11 12**

**monthDay=[31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]**

**r = monthDay[m-1]**

**if m==2:**

**return r + leapYear(y) #处理闰年的2月天数**

**else:**

**r=r**

**\_\_\_\_\_\_\_\_\_\_**

**#计算从公元1年1月1日到 y年m月d日的天数(含两端的函数)**

**def calcDays(y,m,d):**

**if y<1:y=1**

**if m<1:m=1**

**if m>12:m=12**

**if d<1:d=1**

**if d>getLastDay(y,m):**

**d=getLastDay(y,m)**

**count = 0**

**for i in range(1, y):**

**count = count \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**for i in range(1, m):**

**count = count + getLastDay(y, i)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**y,m,d = eval(input("请输入要计算的年月日:"))**

**days = calcDays(y,m,d)**

**print( "从1年1月1日到",y,"年",m,"月",d,"日 共", days, "天")**

**七、插入排序算法是一种简单直观的排序算法。它的工作原理是通过构建有序序列，对于未排序数据，在已排序序列中从后向前扫描，找到相应位置并插入。请找出BUG，运行截图。给出自己对插入排序算法、以及：“程序=算法+数据结构”的理解。**

**其算法描述如下：**

**一般来说，插入排序都采用in-place在数组上实现。具体算法描述如下：**

1. **从第一个元素开始，该元素可以认为已经被排序；**
2. **取出下一个元素，在已经排序的元素序列中从后向前扫描；**
3. **如果该元素（已排序）大于新元素，将该元素移到下一位置；**
4. **重复步骤3，直到找到已排序的元素小于或者等于新元素的位置；**
5. **将新元素插入到该位置后；**
6. **重复步骤2~5。**

**其python程序编写如下：**

**def insertion\_sort(arr):**

**for i in range(len(arr)):**

**# 获取当前需要排序的元素**

**key = arr[i]**

**# 从已排序的序列的末尾开始比较，并找到插入点**

**j = i - 1**

**while j >= 0 and key < arr[j]:**

**# 如果发现已排序的序列中有比当前元素更小的，则交换它们**

**arr[j + 1] = arr[j]**

**arr[j] = key**

**j -= 1**

**return arr**

**# 示例使用**

**arr = [12, 11, 13, 5, 6]**

**sorted\_arr = insertion\_sort(arr)**

**print("Sorted array:", sorted\_arr)**