****Python第一次试验报告——Turtle库****

****题目①——鸡兔同笼****

****一、源程序调试过程****

1. #在使用split进行切片多项输入时,切片标志要选择正确：

|  |
| --- |
| 1. #iHeads,iFeet = map(int,input().split("")) #split切片方法中标识符错误 应该改成空格 2. iHeads,iFeet = map(int,input().split(" ")) |

2. #由于在输入时使用了split切片方法, 元组解包时将输入的iHeaders,iFeet解析为了字符串形;后用iHeaders,iFeet参与运算时使用了除法（/2）,字符串形解包为浮点型，此处直接用format应用与 插槽限制整形{:d}矛盾，使用前需要转为整形int才能输出

|  |
| --- |
| 1. #print("{:d} {:d}".format(iChicken,iRabbits)) 浮点型与插槽限制整形{:d}矛盾需要转为整形 2. print("{:d} {:d}".format(int(iChicken),int(iRabbits))) |

3.#此处在书写异常条件时要注意连个地方,一是边界调节的判断,有的地方可以取等,二不能取等:

|  |
| --- |
| 1. if iRabbits>=0 and iChicken>=0 2. if iRabbits\*iChicken>=0: #此处可以对判断条件简化 降低空间复杂度 |

****二、实验实习结果分析****  
1、运行鸡兔同笼完整运算代码

|  |
| --- |
| 1. #jttl.py 2. iHeads,iFeet = map(int,input().split(" ")) #同时输入头数和脚数，map()方法是一个遍历方法 3. iRabbits = (iFeet - 2 \* iHeads) /2 #使用公式计算兔数，注意这里的iRabbits使浮点型 4. iChicken = iHeads - iRabbits #使用公式计算鸡数 5. if iRabbits\*iChicken>=0: #鸡和兔的数量应同时非负 6. print("{:d} {:d}".format(int(iChicken),int(iRabbits))) #同时输出鸡和兔的数量 7. elif iRabbits\*iChicken<0: #鸡和兔的数量为负时 8. print("Data Error!") #应该输出错误 |

****三、心得体会****

****题目②——百分制转五分制****

1. ****源程序调试过程****  
   1、精度保证

|  |
| --- |
| 1. #grade = input() 由于题目输入条件要求输入可能为浮点型,为了保证精度用浮点型声明 2. grade = float(input()) |

2、区间需要闭合

|  |
| --- |
| 1. #elif 0<grade<=59: 此处区间需要完整覆盖要求的区间,不要掉了等号 2. #elif 0=<grade<=59: 这种写法是非法的 只有<= 和>= 3. elif 0<=grade<=59: |

****二、实验实习结果分析****

|  |
| --- |
| 1. #grade.py 2. grade = float(input()) #输入一个可能为浮点数的数字 3. if not 0<=grade<=100: #判断输入是否非法（百分制） 4. print("Not valid") 5. else: #使用多分支语句对分数进行分段 赋为五分制 6. if 89<grade<=100: 7. print("A") 8. elif 79<grade<=89: 9. print("B") 10. elif 69<grade<=79: 11. print("C") 12. elif 59<grade<=69: 13. print("D") 14. elif 0<=grade<=59: 15. print("E") |

****三、心得体会****  
（通过该实验，你学到了什么？）