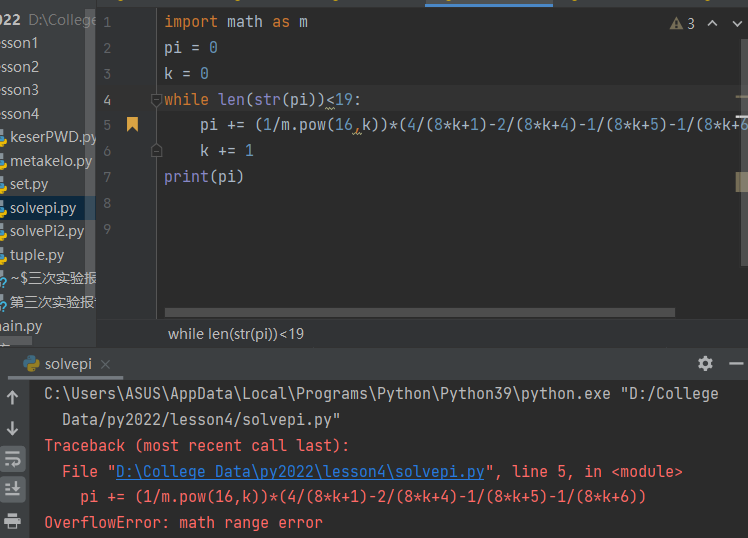
****Python第三次试验报告****

****题目①——门特卡罗法求圆周率****

****一、源程序调试过程****

1. #直接使用圆周率公式进行迭代计算,迭代次数超过18次会出现内存溢出,此时运算精度达不到要求。

|  |
| --- |
| 1. import math as m 2. pi = 0 3. k = 0 4. while len(str(pi))<1: 5. pi += (1/m.pow(16,k))\*(4/(8\*k+1)-2/(8\*k+4)-1/(8\*k+5)-1/(8\*k+6)) 6. k += 1 7. print(pi) |



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、运行圆周率计算完整代码

|  |
| --- |
| 1. import random #使用随机种子需要全局引入 2. from math import sqrt as Sqrt #引入math库中的开方函数 3. from time import process\_time as Clk #引入计时器 4. DARTS = int(input()) #输入一个样本数字 5. hits = 0.0 6. random.seed(123) #设定随机数种子 7. Clk() #开始记时 8. for i in range(1,DARTS+1): 9. x,y = random.random(),random.random() #生成随机坐标 10. dist = Sqrt(x \*\* 2 + y \*\* 2) #计算随机生成坐标点到圆心的距离为 11. if dist <= 1.0: #限制随机生成坐标点离圆心距离为2（限制在圆内） 12. hits += 1 #随机点计数器 13. pi = 4 \* (hits/DARTS) #根据圆形面积pi\*R^2与正方形面积R^2 四分之一圆形与4分之一正方形 的比例为 (pi\*R^2)/4 / 0.5R\*0.5R = pi/4 14. print("{:.6f}".format(pi)) #打印圆周率结果 15. # print("Time is {:5f}s".format(Clk())) #打印程序运行耗时 |

****三、心得体会****

****题目②——凯撒加密****

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****

|  |
| --- |
| # C = (P+n) mod 26  bitMove,Str = input().split(' ',1) #使用split函数对输入字符串进行元组解包，在第一个空格位置切片  bitMove = int(bitMove) #输入时默认经过切片后输出的变量为诶字符型 需要转为整形才能才能与运算  for p in Str: #遍历输入数组的所有元素  if "a" <= p <= "z": #对所有小写字母做凯撒加密  print(chr(ord("a") +(ord(p) - ord("a") + bitMove) % 26), end="")  elif "A"<= p <="Z": #对所有大写字母做凯撒加密  print(chr(ord("A")+(ord(p) - ord("A")+ bitMove) % 26),end="")  else:  print(p,end="") #对其他情况如输入数字空格保持原型不修改 |

****三、心得体会****  
（通过该实验，你学到了什么？）