第十五周Python实验报告（周三）

**学号**：117060400110 **姓名**：蒙柳双

**班级**：17应用统计学1班 **指导老师**：林卫中老师

**实验目的：**

1. 要我们对random库熟悉使用
2. 通过解这些题训练我们的计算思维
3. 对eval库的熟悉使用
4. 对函数的打包
5. 读懂代码

**实验要求：**

1. 上课不玩手机
2. 要认真编码并且思考

3.实在不会的就向老师求助

**实验内容步骤**：

1. 程序练习ex8-1.兵乓球比赛规则如下：每人发两个球，轮换发球权，胜方得分（无论是否为发球方），先得11分者获胜（无论是否为发球方）

**核心代码：**

from random import random

def printIntro():

print("这个程序模拟两个选手A和B的某种竞技比赛")

print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

a = eval(input("请输入选手A的能力值(0-1): "))

b = eval(input("请输入选手B的能力值(0-1): "))

n = eval(input("模拟比赛的场次: "))

return a, b, n

def simNGames(n, probA, probB):

winsA, winsB = 0, 0

for i in range(n):

scoreA, scoreB = simOneGame(probA, probB)

if scoreA > scoreB:

winsA += 1

else:

winsB += 1

return winsA, winsB

def gameOver(a,b):

return a==11 or b==11

def simOneGame(probA, probB):

scoreA, scoreB = 0, 0

serving = 0#

t = 0

while not gameOver(scoreA, scoreB):

if serving == "A":

if random() < probA:

scoreA += 1

else:

#serving="B"#转换球方

scoreB += 1

else:

if random() < probB:

scoreB += 1

else:

#serving="A"

scoreA += 1

t += 1

if t%2 == 0:

serving = (serving+1)%2

return scoreA, scoreB

def printSummary(winsA, winsB):

n = winsA + winsB

print("竞技分析开始，共模拟{}场比赛".format(n))

print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

printIntro()

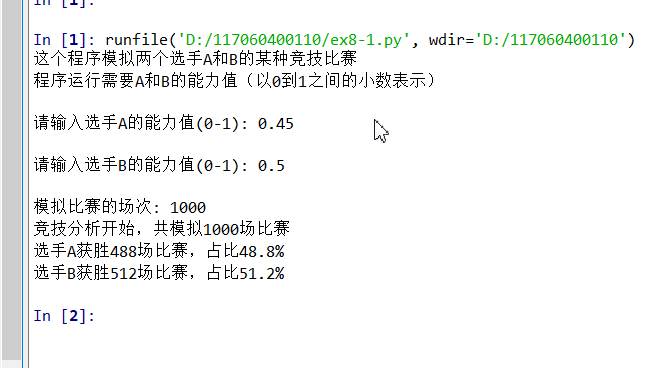
probA, probB, n = getInputs()

winsA, winsB = simNGames(n, probA, probB)

printSummary(winsA, winsB)

main()

**实验结果：**



1. 程序练习ex8-1.兵乓球比赛规则如下：每人发五个球，轮换发球权，胜方得分（无论是否为发球方），先得21分者获胜（无论是否为发球方）

**核心代码：**

from random import random

def printIntro():

print("这个程序模拟两个选手A和B的某种竞技比赛")

print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

a = eval(input("请输入选手A的能力值(0-1): "))

b = eval(input("请输入选手B的能力值(0-1): "))

n = eval(input("模拟比赛的场次: "))

return a, b, n

def simNGames(n, probA, probB):

winsA, winsB = 0, 0

for i in range(n):

scoreA, scoreB = simOneGame(probA, probB)

if scoreA > scoreB:

winsA += 1

else:

winsB += 1

return winsA, winsB

def gameOver(a,b):

return a==21 or b==21

def simOneGame(probA, probB):

scoreA, scoreB = 0, 0

serving = 0#

t = 0

while not gameOver(scoreA, scoreB):

if serving == "A":

if random() < probA:

scoreA += 1

else:

#serving="B"#转换球方

scoreB += 1

else:

if random() < probB:

scoreB += 1

else:

#serving="A"

scoreA += 1

t += 1

if t%2 == 0:

serving = (serving+1)%5

return scoreA, scoreB

def printSummary(winsA, winsB):

n = winsA + winsB

print("竞技分析开始，共模拟{}场比赛".format(n))

print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

printIntro()

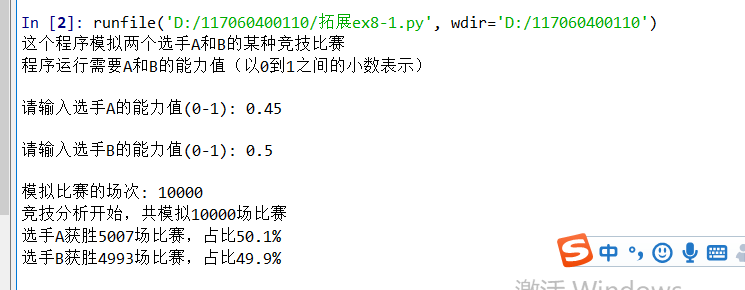
probA, probB, n = getInputs()

winsA, winsB = simNGames(n, probA, probB)

printSummary(winsA, winsB)

main()

**实验结果：**



3.程序练习ex8-1.兵乓球比赛规则如下：，每人发两个球，轮换发球权，胜方得分（无论是否为发球方），先得11分者获胜（无论是否为发球方），如果双方都达10分，则多对方两分的选手获胜

**核心代码：**

from random import random

def printIntro():

print("这个程序模拟两个选手A和B的某种竞技比赛")

print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

a = eval(input("请输入选手A的能力值(0-1): "))

b = eval(input("请输入选手B的能力值(0-1): "))

n = eval(input("模拟比赛的场次: "))

return a, b, n

def simNGames(n, probA, probB):

winsA, winsB = 0, 0

for i in range(n):

scoreA, scoreB = simOneGame(probA, probB)

if scoreA > scoreB:

winsA += 1

else:

winsB += 1

return winsA, winsB

def gameOver(a,b):

"""if (a==11 and b<10) or (a<10 and b==11):

return True

if (a >= 10 and b-a==2) or (b>=10 and a-b==2):

return True

return False"""

if a < b:

a,b = b,a

if a==11 and b<10:

return True

if b>=10 and a-b==2:

return True

return False

def simOneGame(probA, probB):

scoreA, scoreB = 0, 0

serving = 0#

t = 0

while not gameOver(scoreA, scoreB):

if serving == "A":

if random() < probA:

scoreA += 1

else:

scoreB += 1

else:

if random() < probB:

scoreB += 1

else:

scoreA += 1

t += 1

if t%2 == 0:

serving = (serving+1)%2

return scoreA, scoreB

def printSummary(winsA, winsB):

n = winsA + winsB

print("竞技分析开始，共模拟{}场比赛".format(n))

print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

printIntro()

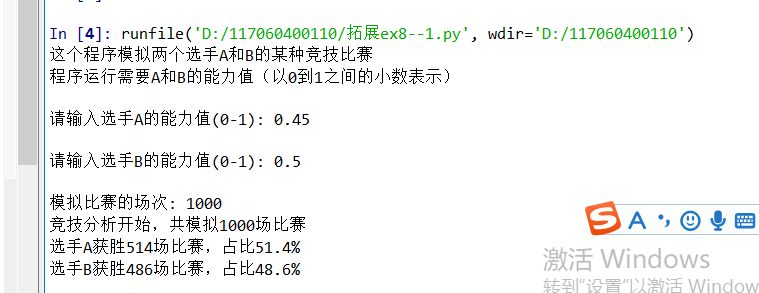
probA, probB, n = getInputs()

winsA, winsB = simNGames(n, probA, probB)

printSummary(winsA, winsB)

main()

**实验结果：**



4.程序练习8-2

规则：甲乙两队比赛，考虑双方的投篮得分和篮板球能力水平。甲队先获发球权，如果甲投篮中，则得1分，发球权交换乙队；如果甲投篮不中，双方抢篮板球，抢得篮板球一方获得球权。每队进攻限时24秒，若超过24秒则交换球权给对方，每局限时比赛12分钟。

**核心代码：**

from random import random

from random import randint

def printIntro():

print("这个程序模拟两个选手A和B的某种竞技比赛")

print("程序运行需要A和B的能力值（以0到1之间的小数表示）")

def getInputs():

a1 = eval(input("请输入选手甲的投篮得分能力值(0-1): "))

b1 = eval(input("请输入选手甲的篮板球能力水平："))

a2 = eval(input("请输入选手乙的投篮得分能力值(0-1)： "))

b2 = eval(input("请输入选手乙的篮板球能力水平："))

n = eval(input("模拟比赛的场次: "))

return a1,b2, a2,b2, n

def simNGames(n,golaA,boarA,golaB,boarB):

winsA, winsB = 0, 0

for i in range(n):

scoreA,scoreB = simOneGame(golaA,boarA,golaB,boarB)

if scoreA > scoreB:

winsA += 1

else:

winsB += 1

return winsA, winsB

def gameOver(t):

return t > 720

def simOneGame(golaA,boarA,golaB,boarB):

scoreA, scoreB = 0, 0

serving = 0

totaltime = 0

while not gameOver(totaltime):

t = randint(1,24)#表示随机产生从1到24中的随机整数

totaltime += t

if t == 24:

serving =(serving + 1)%2

else:

if serving ==0:

if random() < golaA:

scoreA += 1

scoreB = 1

else:

if random() < boarA:

seerving = 0

else:

serving =1

else:

if random()<golaB:

scoreB += 1

scoreA = 1

else:

if random() < boarB:

seerving = 1

else:

serving = 0

return scoreA, scoreB

def printSummary(winsA, winsB):

n = winsA + winsB

print("竞技分析开始，共模拟{}场比赛".format(n))

print("选手A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

print("选手B获胜{}场比赛，占比{:0.1%}".format(winsB, winsB/n))

def main():

printIntro()

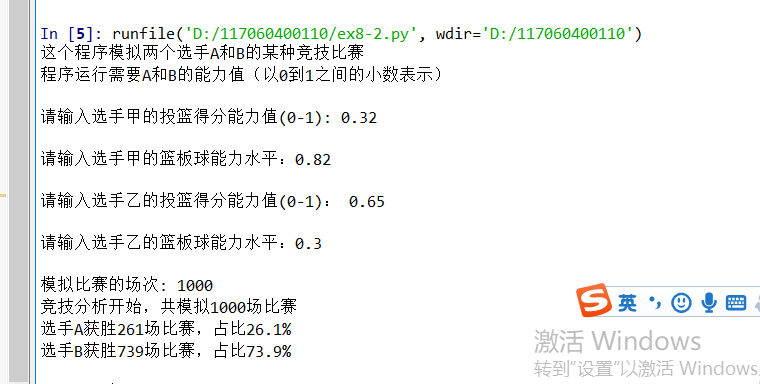
golaA,boarA,golaB,boarB,n = getInputs()

winsA, winsB = simNGames(n,golaA,boarA,golaB,boarB)

printSummary(winsA, winsB)

main()

**实验结果：**



实验总结：

1. 自顶行下式设计是逐步求精的过程，它有一下的步骤：

步骤一：打印程序的介绍信息

步骤二：获得程序运行需要的参数，即probA,proB,n

步骤三：利用球员A和B的能力值probA和probB，模拟n次比赛

步骤四：输出球员A和B获胜比赛的场次和概率

1. 其实我们要明白main()函数是设计分析顶层结构，printIntro()输出程旭介绍，simNGames()函数模拟n场比赛，并追踪每个球员获胜场数，sinOneGame()函数模拟一场比赛，用于知道每个球员的概率，返回两个球员的最终得分，gameOver()函数是用来表示一场比赛结束的条件，printSummary()函数适用于最后结果的输出
2. 通过增加各种条件来训练自己的计算思维
3. 并且通过已经写的、了解的代码进行举一反三