**体育竞技分析**

**学号：** 117060400113 **姓名**： 曹晓静

**班级：** 17应用统计一班  **指导老师：** 林卫中

**一、实验名称**： 计算机程序设计

**二、实验题目：**

**1、体育竞技分析（乒乓球）**

**2、体育竞技分析（篮球）**

**四、算法实现：**

**1、**

**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 == 0:**

**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():**

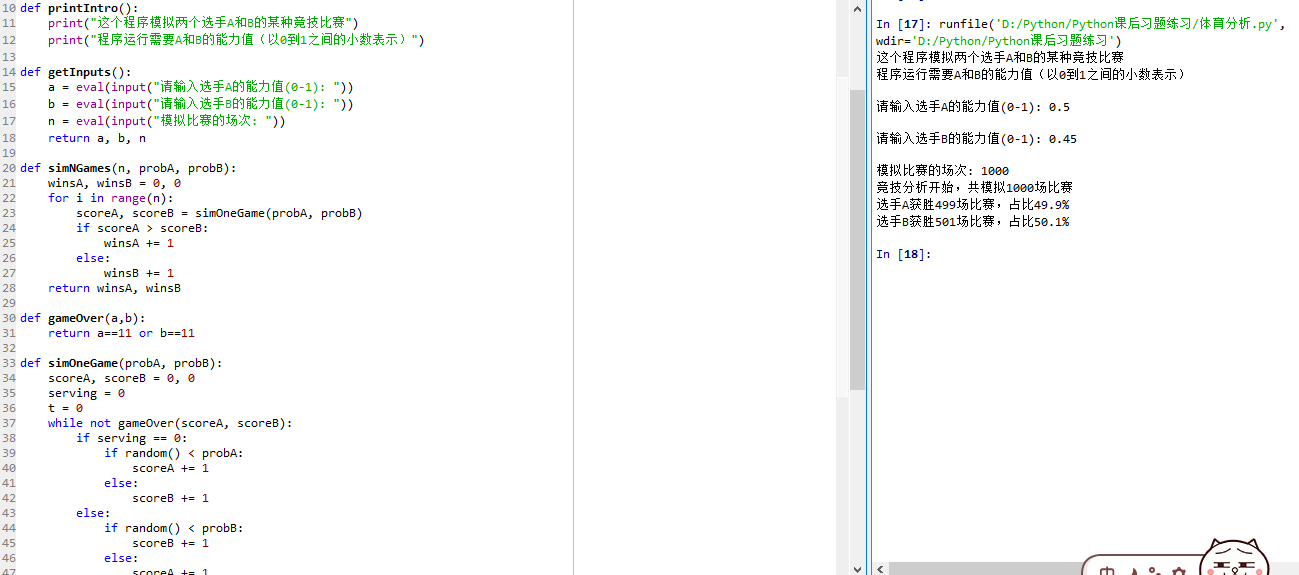
**printIntro()**

**probA, probB, n = getInputs()**

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

**printSummary(winsA, winsB)**

**main()**



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 == 10 and b== 10:

return a-b==2 or b-a==2

else:

return a==11 or b==11

def simOneGame(probA, probB):

scoreA, scoreB = 0, 0

serving = 0#0:表示A发球 1：表示B发球

t = 0

while not gameOver(scoreA, scoreB):

if serving == 0:

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():

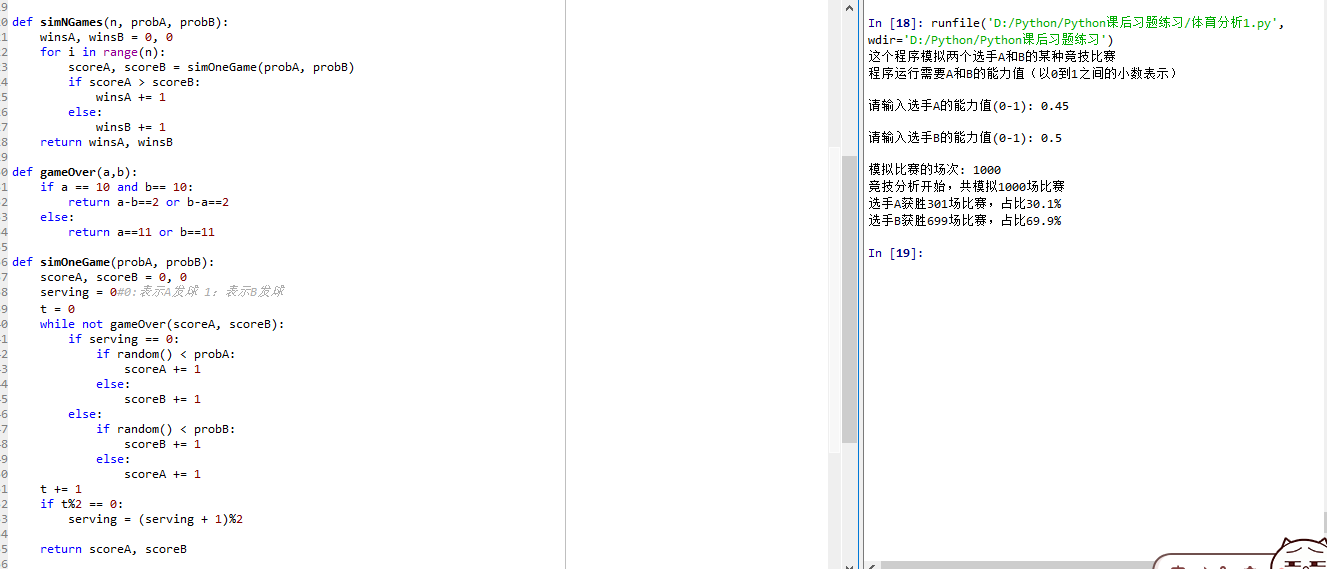
printIntro()

probA, probB, n = getInputs()

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

printSummary(winsA, winsB)

main()



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 (b>=10 and a-b==2) or (a >=10 and b-a==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 == 0:

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():

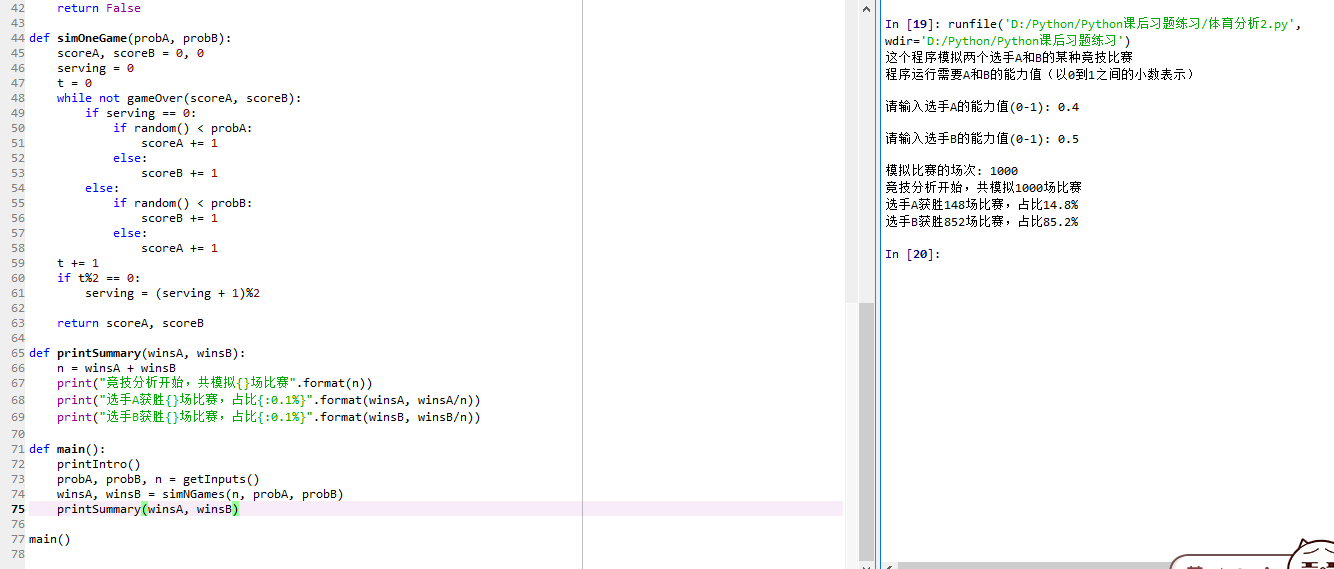
printIntro()

probA, probB, n = getInputs()

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

printSummary(winsA, winsB)

main()



2、

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():

b1 = eval(input("请输入球队A的抢板能力值(0-1): "))

b2 = eval(input("请输入球队B的抢板能力值(0-1): "))

g1 = eval(input("请输入球队A的投篮能力值(0-1): "))

g2 = eval(input("请输入球队B的投篮能力值(0-1): "))

n = eval(input("模拟比赛的场次: "))

return b1, b2, g1, g2, n

def simNGames(n, goleA, boardA, goleB, boardB):

winsA, winsB = 0, 0

for i in range(n):

scoreA, scoreB = simOneGame(goleA, boardA, goleB, boardB)

if scoreA > scoreB:

winsA += 1

else:

winsB += 1

return winsA, winsB

def gameOver(t):

return t >= 12\*60

def simOneGame(goleA, boardA, goleB, boardB):

scoreA, scoreB = 0, 0

serving = 0

totalTime = 0

while not gameOver(totalTime):

t = randint(1, 24)

totalTime += t

if t == 24:

serving = (serving + 1)%2

else:

if serving == 0:

if random() < goleA:

scoreA += 1

serving = 1

else:

if random() < boardA:

serving = 0

else:

serving = 1

else:

if random() < goleB:

scoreB += 1

serving = 0

else:

if random() < boardB:

serving = 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("球队A获胜{}场比赛，占比{:0.1%}".format(winsA, winsA/n))

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

def main():

printIntro()

boardA, boardB, goleA, goleB, n = getInputs()

winsA, winsB = simNGames(n, goleA, boardA, goleB, boardB)

printSummary(winsA, winsB)

main()

