**实验报告15**

**学号：**117060400123 **姓名**：黄茜洋 **班级：**应统一班**指导老师：**林卫中**实验名称**：第八章程序设计方法论

**实验要求：**（1）了解计算思维的概念

（2）掌握自顶向下的设计方法

（3）掌握自底向上的执行过程

**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==21 or b==21

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%5 == 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():

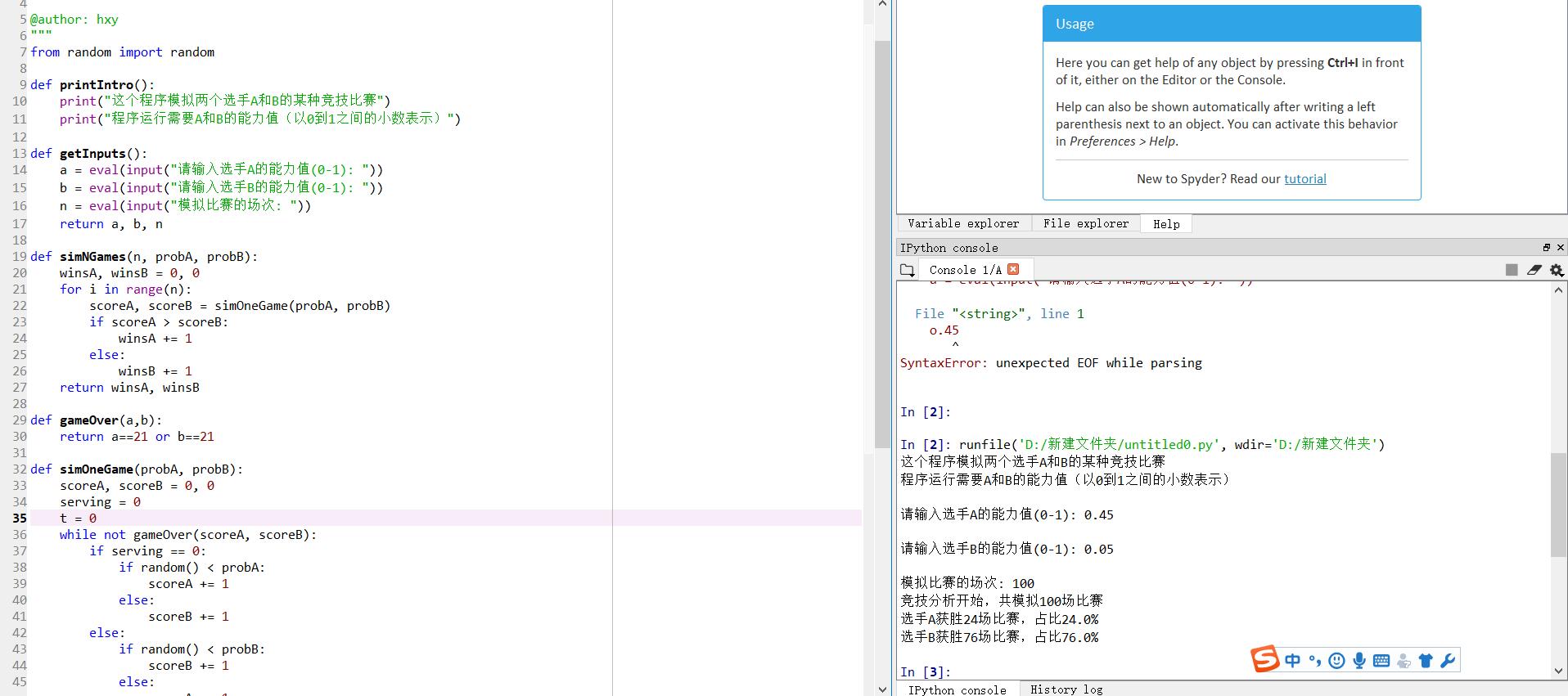
printIntro()

probA, probB, n = getInputs()

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

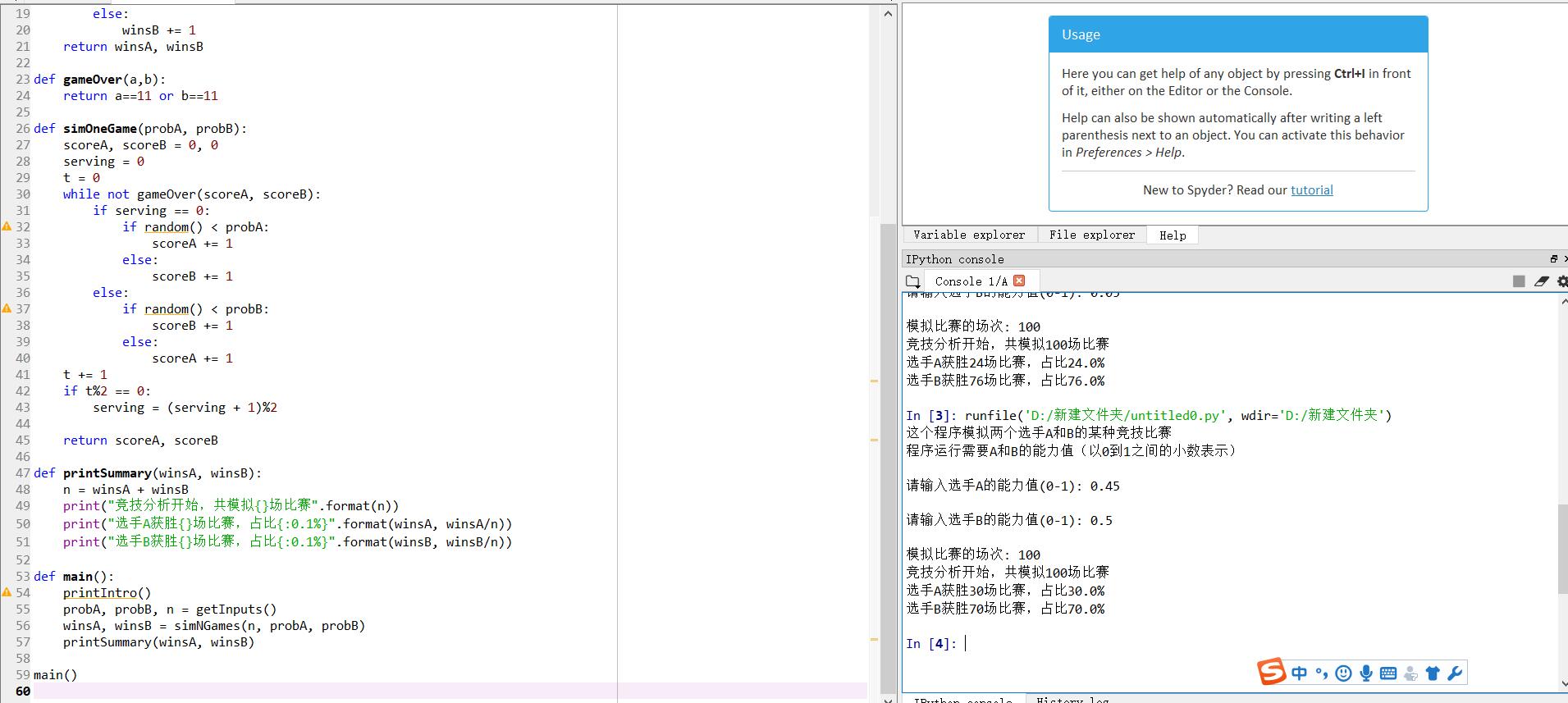
printSummary(winsA, winsB)

main()



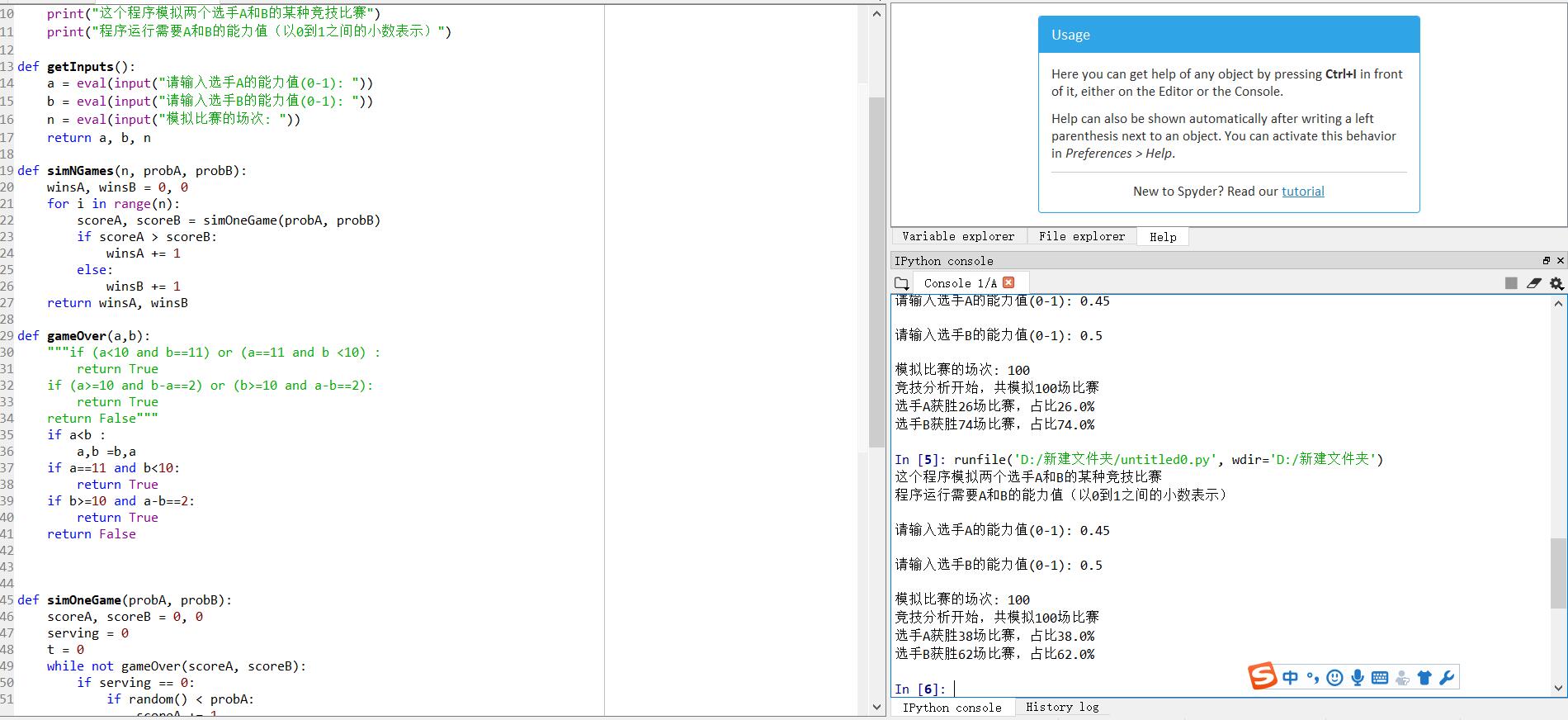
2.

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, ndef 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, winsBdef gameOver(a,b): return a==11 or b==11def 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, scoreBdef 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()



3.

from random import randomdef 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, ndef 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, winsBdef gameOver(a,b): if (a<10 and b==11) or (a==11 and b <10) : return True if (a>=10 and b-a==2) or (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, scoreBdef 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()4.

from random import randomdef 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, ndef 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, winsBdef gameOver(a,b): """if (a<10 and b==11) or (a==11 and b <10) : 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 == 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, scoreBdef 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()

5.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():

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

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

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

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

n = eval(input("模拟比赛的场次: "))

return g1, b1, g2, b2, 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 #0: 代表A队发球，1：代表B队发球

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("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():

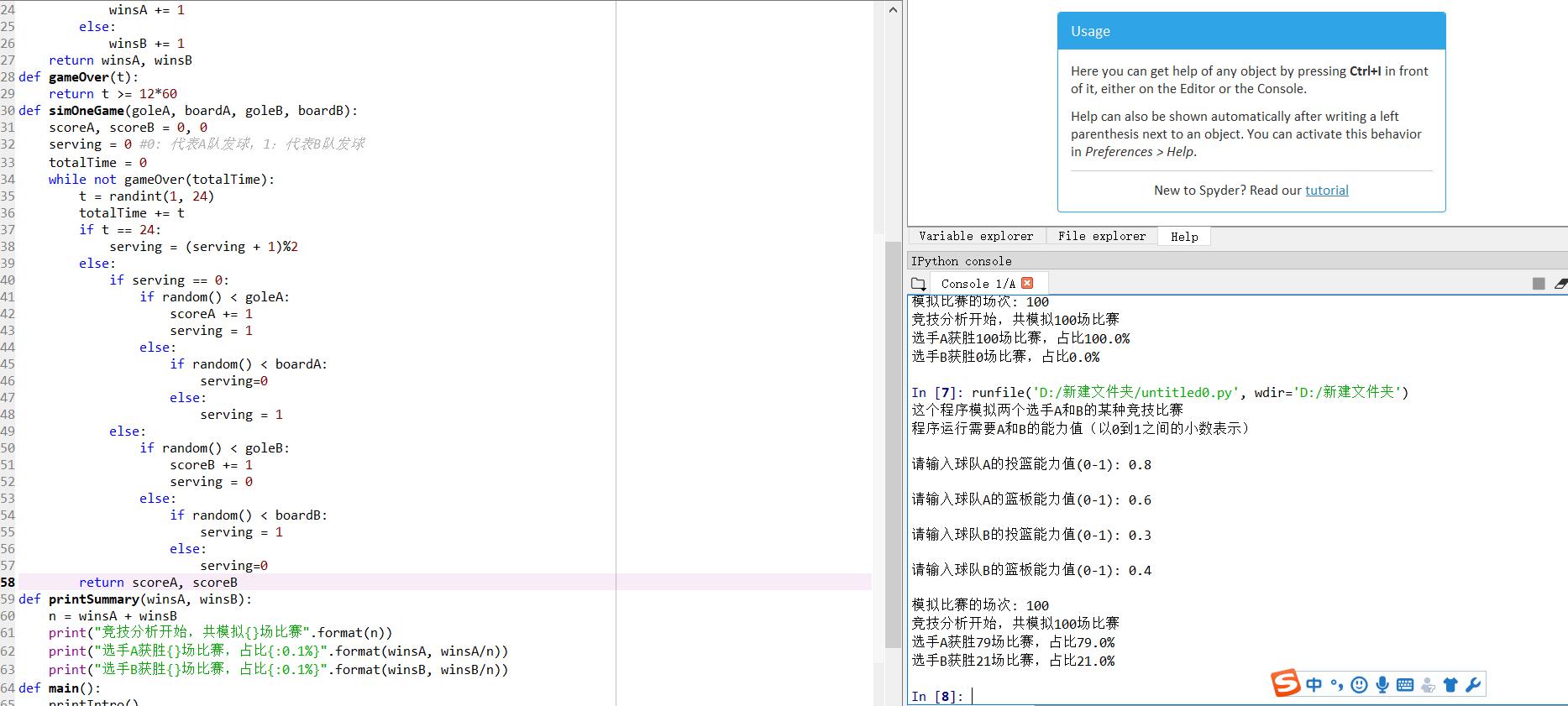
printIntro()

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

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

printSummary(winsA, winsB)

main()



**实验总结：**

经过这次实验，我学会了如何用python去模拟游戏规则，自顶向下的设计方法。