**CMPSC 121 Project Report**

**Project #2**

**Sdh5378 – Shane Hagan**

**03/17/17**

**Code**

**Full Code:**

# Project 2

import random

# Tell the user how many people are dining

print("For this simulation, we will say there are five people dining in this party.")

# Server Efficiency Constant

SEC = 2

# Randomization

# 1. Random Number

a = 30

AandS = int(a \* random.random()) + 1

print("1. Arrival and Seating: Let's say this party is seated",AandS,"minutes after arriving.\n")

# 2. Seating Constant

# We will say the constant here is 3 minutes

SConst = 3

print("2. Seating: It takes",SConst,"minutes for the host to seat this party.\n")

# 3. Server Arrival and Ordering Constants

SAOC = 5

print('''3. Server Arrival and Ordering: Say the \"server efficiency constant\" is 2 minutes i.e. it takes 2 minutes everytime we need to wait for the server to do something.

We wait 2 minutes for the server to arrive. Then, suppose we have 5 people in this party and the time it takes each person to order is

0.60 minutes. Thus, we waited 2 minutes for the server to arrive and 3 minutes for everyone to order, so this step took 5 total minutes.\n''')

# 4. Wait for food randomizing

b = 25

P1 = int(b \* random.random()) + 1

P2 = int(b \* random.random()) + 1

P3 = int(b \* random.random()) + 1

P4 = int(b \* random.random()) + 1

P5 = int(b \* random.random()) + 1

FinalWait = max(P1, P2, P3, P4, P5)

print('''4. Wait for Food: Suppose it takes''',P1,'''minutes for the first person's meal,''',P2,'''minutes for the second person's meal,''',P3,'''minutes for the third person's meal,''',P4,'''minutes for the fourth person's meal, and''',P5,'''minutes for the fifth person's meal. Thus, it will take''',FinalWait,'''

minutes until the entire party's meal is ready to be served.\n''')

# 5. Serving of food

print("5. Serving of Food: The \"server efficiency constant\" of 2 minutes must be waited until the party gets its food.\n")

# 6 Eating Time Randomization

c = 45

EatTime = int(c \* random.random()) + 1

print("6. Eating Time: Suppose it takes this party",EatTime,"minutes to eat their meal.\n")

# 7 Dessert Ordering Randomization

x = 1

y = 5

Dessert = random.randint(x,y)

DessertOrder = Dessert \* 0.60

print('''7. Dessert Ordering: Suppose''',Dessert,'''of''',y,'''people in this party want dessert. Thus it will

take each person 0.60 minutes (the constant from above) and a total of''',DessertOrder,'''minutes

to order their desserts.\n''')

# 8. Wait for Dessert Randomization

d = 7

# Add if statement corresponding to if there is anyone who wants dessert

if Dessert == 1:

D1 = int(d \* random.random()) + 1

DMax = D1

print('''8. Wait For Dessert: Since one person ordered dessert, we only wait for their

dessert to be prepared. We will wait''',DMax,''' minutes''')

elif Dessert == 2:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

DMax = max(D1,D2)

print('''8. Wait For Dessert: Suppose the first person's dessert takes''',D1,'''minutes to prepare

and the second person's dessert takes''',D2,'''minutes to prepare. Thus, we wait''',DMax,'''

minutes for the dessert to arrive.\n''')

elif Dessert == 3:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3)

print('''8. Wait For Dessert: Suppose the first person's dessert takes''',D1,'''minutes to prepare,

the second person's dessert takes''',D2,'''minutes to prepare, and the third person's

dessert takes''',D3,'''minutes to prepare. Thus, we wait''',DMax,'''minutes for the dessert to arrive.\n''')

elif Dessert == 4:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4)

print('''8. Wait For Dessert: Suppose the first person's dessert takes''',D1,'''minutes to prepare,

the second person's dessert takes''',D2,'''minutes to prepare, the third person's

dessert takes''',D3,'''minutes to prepare, and the fourth person's dessert takes''',D4,'''minutes

to prepare. Thus, we wait''',DMax,'''minutes for the dessert to arrive.\n''')

elif Dessert == 5:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

D5 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4,D5)

print('''8. Wait For Dessert: Suppose the first person's dessert takes''',D1,'''minutes to prepare,

the second person's dessert takes''',D2,'''minutes to prepare, the third person's

dessert takes''',D3,'''minutes to prepare, the fourth person's dessert takes''',D4,'''minutes

to prepare, and the fifth person's dessert takes''',D5,'''minutes to prepare.

Thus, we wait''',DMax,'''minutes for the dessert to arrive.\n''')

# 9. Serving of Dessert

print("9. Serving of Dessert: We wait 2 minutes for the dessert to arrive.\n")

# 10. End of Meal Randomization

e = 30

End = int(e \* random.random()) + 1

print("10. End of Meal: Suppose this group stays at the table",End,"minutes.\n")

Total = AandS + SConst + SAOC + FinalWait + SEC + EatTime + DessertOrder + DMax + SEC + End

print('''Thus, it took''',AandS,'''+''',SConst,'''+''',SAOC,'''+''',FinalWait,'''+''',SEC,'''+''',EatTime,'''+''',DessertOrder,'''+''',DMax,'''+''',SEC,'''+''',End,'''=''',Total,'''minutes from when

this party arrived until it left.''')

**Numerical Mode:**

# Project 2

import random

# Tell the user how many people are dining

print("For this simulation, we will say there are five people dining in this party.")

# User Input

Parties = int(input("Enter the number of parties: "))

t = 1

Total\_Time = 0

while t < Parties + 1:

# Server Efficiency Constant

SEC = 2

# Randomization

# 1. Random Number

a = 30

AandS = int(a \* random.random()) + 1

# 2. Seating Constant

# We will say the constant here is 3 minutes

SConst = 3

# 3. Server Arrival and Ordering Constants

SAOC = 5

# 4. Wait for food randomizing

b = 25

P1 = int(b \* random.random()) + 1

P2 = int(b \* random.random()) + 1

P3 = int(b \* random.random()) + 1

P4 = int(b \* random.random()) + 1

P5 = int(b \* random.random()) + 1

FinalWait = max(P1, P2, P3, P4, P5)

# 5. Serving of food

# 6 Eating Time Randomization

c = 45

EatTime = int(c \* random.random()) + 1

# 7 Dessert Ordering Randomization

x = 1

y = 5

Dessert = random.randint(x,y)

DessertOrder = Dessert \* 0.60

# 8. Wait for Dessert Randomization

d = 7

# Add if statement corresponding to if there is anyone who wants dessert

if Dessert == 1:

D1 = int(d \* random.random()) + 1

DMax = D1

elif Dessert == 2:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

DMax = max(D1,D2)

elif Dessert == 3:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3)

elif Dessert == 4:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4)

elif Dessert == 5:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

D5 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4,D5)

# 9. Serving of Dessert

# 10. End of Meal Randomization

e = 30

End = int(e \* random.random()) + 1

Total = AandS + SConst + SAOC + FinalWait + SEC + EatTime + DessertOrder + DMax + SEC + End

Total\_Time += Total

print("Trial",t,":",Total,"minutes")

t += 1

AvgTime = Total\_Time / Parties

print("Average dining time among all parties in the simulation:",AvgTime,"minutes")

**Plotting Mode:**

# Project 2

import random

# Tell the user how many people are dining

print("For this simulation, we will say there are five people dining in this party.")

# User Input

Parties = int(input("Enter the number of parties: "))

t = 1

s = "\*"

Total\_Time = 0

while t < Parties + 1:

# Server Efficiency Constant

SEC = 2

# Randomization

# 1. Random Number

a = 30

AandS = int(a \* random.random()) + 1

# 2. Seating Constant

# We will say the constant here is 3 minutes

SConst = 3

# 3. Server Arrival and Ordering Constants

SAOC = 5

# 4. Wait for food randomizing

b = 25

P1 = int(b \* random.random()) + 1

P2 = int(b \* random.random()) + 1

P3 = int(b \* random.random()) + 1

P4 = int(b \* random.random()) + 1

P5 = int(b \* random.random()) + 1

FinalWait = max(P1, P2, P3, P4, P5)

# 5. Serving of food

# 6 Eating Time Randomization

c = 45

EatTime = int(c \* random.random()) + 1

# 7 Dessert Ordering Randomization

x = 1

y = 5

Dessert = random.randint(x,y)

DessertOrder = Dessert \* 0.60

# 8. Wait for Dessert Randomization

d = 7

# Add if statement corresponding to if there is anyone who wants dessert

if Dessert == 1:

D1 = int(d \* random.random()) + 1

DMax = D1

elif Dessert == 2:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

DMax = max(D1,D2)

elif Dessert == 3:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3)

elif Dessert == 4:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4)

elif Dessert == 5:

D1 = int(d \* random.random()) + 1

D2 = int(d \* random.random()) + 1

D3 = int(d \* random.random()) + 1

D4 = int(d \* random.random()) + 1

D5 = int(d \* random.random()) + 1

DMax = max(D1,D2,D3,D4,D5)

# 9. Serving of Dessert

# 10. End of Meal Randomization

e = 30

End = int(e \* random.random()) + 1

Total = AandS + SConst + SAOC + FinalWait + SEC + EatTime + DessertOrder + DMax + SEC + End

Total2 = int(Total)

# Create the output of stars

Star = (Total2 - 60) % 5

Remand = 5 - Star

Total3 = Total2 + Remand

if Star > 0:

StarFinal = int((Total3-60) / 5)

elif Star == 0:

StarFinal = int((Total2 - 60) / 5)

OutPut = StarFinal \* s

Total\_Time += Total

print("Trial",t,":",OutPut)

t += 1

AvgTime = Total\_Time / Parties

print("Average dining time among all parties in the simulation:",AvgTime,"minutes")

**Sample Runs**

**One sample run, 50 trials (Numeric):**

For this simulation, we will say there are five people dining in this party.

Enter the number of parties: 50

Trial 1 : 121.2 minutes

Trial 2 : 69.0 minutes

Trial 3 : 75.0 minutes

Trial 4 : 82.2 minutes

Trial 5 : 96.0 minutes

Trial 6 : 76.4 minutes

Trial 7 : 107.2 minutes

Trial 8 : 88.2 minutes

Trial 9 : 97.0 minutes

Trial 10 : 93.8 minutes

Trial 11 : 105.6 minutes

Trial 12 : 107.2 minutes

Trial 13 : 94.4 minutes

Trial 14 : 68.0 minutes

Trial 15 : 79.2 minutes

Trial 16 : 98.8 minutes

Trial 17 : 104.0 minutes

Trial 18 : 91.6 minutes

Trial 19 : 106.8 minutes

Trial 20 : 72.8 minutes

Trial 21 : 89.2 minutes

Trial 22 : 98.4 minutes

Trial 23 : 89.8 minutes

Trial 24 : 89.0 minutes

Trial 25 : 113.8 minutes

Trial 26 : 111.6 minutes

Trial 27 : 98.0 minutes

Trial 28 : 72.2 minutes

Trial 29 : 78.8 minutes

Trial 30 : 57.2 minutes

Trial 31 : 74.4 minutes

Trial 32 : 109.8 minutes

Trial 33 : 80.6 minutes

Trial 34 : 87.8 minutes

Trial 35 : 98.6 minutes

Trial 36 : 103.2 minutes

Trial 37 : 91.0 minutes

Trial 38 : 60.4 minutes

Trial 39 : 77.4 minutes

Trial 40 : 97.2 minutes

Trial 41 : 72.6 minutes

Trial 42 : 72.2 minutes

Trial 43 : 118.0 minutes

Trial 44 : 83.0 minutes

Trial 45 : 85.0 minutes

Trial 46 : 114.6 minutes

Trial 47 : 85.2 minutes

Trial 48 : 100.6 minutes

Trial 49 : 60.0 minutes

Trial 50 : 93.2 minutes

Average dining time among all parties in the simulation: 89.944 minutes

**One sample run, 50 trials (Plotting):**

For this simulation, we will say there are five people dining in this party.

Each "\*" represents 5 minutes past 60 minutes

Enter the number of parties: 50

Trial 1 : \*\*

Trial 2 : \*\*\*

Trial 3 : \*\*\*\*\*\*\*\*

Trial 4 : \*\*\*\*\*\*\*

Trial 5 : \*\*\*\*\*\*\*\*\*\*\*\*\*

Trial 6 : \*

Trial 7 : \*\*\*\*\*\*\*\*

Trial 8 : \*\*\*

Trial 9 : \*\*\*\*\*\*\*\*

Trial 10 : \*\*\*\*\*\*\*\*\*\*\*

Trial 11 : \*

Trial 12 : \*\*\*\*\*\*\*\*\*\*\*\*\*

Trial 13 : \*\*

Trial 14 : \*\*\*\*

Trial 15 : \*\*\*\*\*\*\*\*\*

Trial 16 : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Trial 17 : \*\*\*\*\*\*\*\*\*\*\*\*\*\*

Trial 18 : \*\*\*\*\*\*

Trial 19 : \*\*\*\*\*\*\*

Trial 20 : \*\*\*\*\*\*\*\*

Trial 21 : \*\*\*\*\*\*\*\*

Trial 22 : \*\*\*\*\*\*\*\*

Trial 23 : \*\*\*\*\*\*\*\*

Trial 24 : \*\*\*\*\*\*\*\*\*

Trial 25 : \*\*\*\*\*\*\*

Trial 26 :

Trial 27 : \*\*\*\*

Trial 28 : \*\*\*

Trial 29 : \*\*\*\*\*

Trial 30 : \*\*\*

Trial 31 : \*\*\*\*\*

Trial 32 :

Trial 33 : \*\*\*\*\*\*\*

Trial 34 : \*\*\*\*\*

Trial 35 : \*\*\*\*\*\*\*\*\*

Trial 36 : \*\*\*\*\*\*\*\*\*

Trial 37 : \*\*\*\*\*\*

Trial 38 : \*\*\*\*\*\*\*\*

Trial 39 : \*\*\*\*\*\*

Trial 40 : \*\*\*\*\*\*\*\*\*\*

Trial 41 : \*\*\*\*\*\*\*

Trial 42 : \*\*\*\*\*\*\*\*\*\*

Trial 43 : \*\*\*\*\*\*\*\*

Trial 44 : \*\*\*\*\*\*\*

Trial 45 : \*\*\*\*\*\*\*\*\*

Trial 46 : \*\*\*\*\*\*

Trial 47 : \*\*\*\*

Trial 48 : \*\*\*\*\*\*\*\*

Trial 49 : \*\*\*\*

Trial 50 : \*\*\*\*\*\*\*\*\*

Average dining time among all parties in the simulation: 92.016 minutes

**10 times, 10 trials per run:**

|  |  |
| --- | --- |
| Run # | Average Dining Time |
| 1 | 93.34 min |
| 2 | 87.34 min |
| 3 | 90.56 min |
| 4 | 93.66 min |
| 5 | 90.22 min |
| 6 | 92.92 min |
| 7 | 94.56 min |
| 8 | 90.08 min |
| 9 | 93.10 min |
| 10 | 95.28 min |

**10 times, 250 trials per run:**

|  |  |
| --- | --- |
| Run # | Average Dining Time |
| 1 | 95.36 min |
| 2 | 95.30 min |
| 3 | 95.58 min |
| 4 | 95.77 min |
| 5 | 94.98 min |
| 6 | 93.94 min |
| 7 | 94.69 min |
| 8 | 94.41 min |
| 9 | 93.89 min |
| 10 | 94.75 min |

**Discussion**

I would say in this project I reinforced using if and elif statements. As far as learning anything new, I would say I gained more experience and learned how to work with multiple variables at once, each needing to be adjusted and set carefully because some influenced another. I had issues mainly with obtaining the average at the end, which was easily fixed by making a variable to add up the times, and to get the asterisk to the correct amount. I learned more about formulas and making more variables from other variables. I believe there is some way to shorten the program, maybe use other commands as shortcuts, versus my code now. Overall, I’d say the project went well, there were not many errors, and I understood most of the code and how it all worked. It also worked on the first time through, versus other times when there could be an error or two that need to be fixed and corrected. It was not too hard to adjust the code and make it just print out averages, or even allow user to change the amount of trials. Most of the topics we learned were implemented and reinforced in this project also.