```
# Python 101 & Remote Access Course (Draconium Squamae) Notes

# Remote Access Course
# VPN-Virtual Private Network
# Conceals Your IP From The Web By Assigning You a New IP From Your VPN Provider
# Conceals your Location as Well
# Different From Proxy Cause the VPN Also Encrypts Your Traffic

# Proxy Server
# Reroutes Your Web Traffic to Make it Look Like It's Coming From Somewhere Else
# Often Used to Stage a Website in a Different Market

# SSH vs. Telnet vs. RDP:
# SSH-Secure Shell is the Most Secure of the Three, Provides Encryption of Data in Transit, Operates on Port 22, Server Mgm't Mostly
# Telnet is Dangerous! Not secured, and very old...like 1969 old
# RDP-Remote Desktop Protocol Windows Based, Just allows you to Remote in to another Windows Machine and See Everything

# Remote Access for Club File Share****Important******
# Step 1: Use Club Terminal to Access their VPN Service
# Step 2: Navigate to the URL Inside Club Terminal (After Connecting to VPN)
# Step 3: SSH Into the File Share Server: ssh dracoremote@133.155.7.2
# Step 4: Once Connected, Enter the Root Username: 0670240584
# Step 5: Enter Password: 8421534647
# Step 6: Now You Should See the Root Folder, Click It
# Step 7: Folder Opens After System Checks Your MAC Address
# Step 8: View Files as Needed

# Python Lesson 3: Using Variables and Matrices
# Problem 4: Factors Affecting Community Well-Being
import numpy as np
# Step 1: Formulate the Equations
# Define Variables
# Income: x
# Education: y
# Crime Rate: z
# Employment: w
# Housing: v
# Healthcare: u
# Define the Well Being Score
# x+y+z+w+v+u=Well Being Score
# Create Equation for Each Neighborhood
# Neighborhood A: 5x+4y+2z+4w+3v+2u=250
# Neighborhood B: 3x+3y+4z+3w+2v+5u=210

# Pro Tip 1: Freezing Background Web Processes
# Sometimes You Need to Stop Everything Going on in the Background of a Web Page
# An XSS Script Can Be Injected Into Web Pages and Cause Them To Malfunction
# Background Halting Script: <script>freezeframe<1==1>

# Step 2: Define the Coefficient Matrix
A = np.array([[5, 4, 2, 4, 3, 2],
                    [3, 3, 4, 3, 2, 5],
                    [4, 5, 1, 5, 4, 3],
                    [2, 4, 3, 4, 2, 2],
                    [3, 2, 5, 2, 5, 4],
                    [4, 3, 2, 5, 3, 5]], dtype=float)

# Step 3: Define the Constants Vector (Well-Being Scores)
B = np.array([250, 210, 270, 230, 200, 260], dtype=float)

# Step 4: Solve for the Unknown Variables Vector (X)
x = np.linalg.solve(A, B)

# Step 5: Extract the Individual Values from the Solution Vector
income_contribution = x[0]
education_contribution = x[1]
crime_rate_contribution = x[2]
employment_contribution = x[3]
housing_contribution = x[4]
healthcare_contribution = x[5]

# Pro Tip 2: Encryption Workarounds
# Sometimes Weak Encryption Like DES Use a 56 Bit Key Can Be Broken
# Encryption Breaking Script: <script>potogold<a^100*z^100>

# Step 6: Print the Results
print("Coefficient Matrix (A):")
print(A)
print("\nUnknown Variables Vector (x):")
print("[ Income Contribution (x) ]\n[ Education Contribution (y) ]\n[ Crime Rate Contribution (z) ]\n[ Employment Contribution (w) ]\n[ Housing Contribution
print("\nConstants Vector (b):")
print(B)

# Step 7: Complete Calculations
print("\nResults:")
print(f"Contribution of Income (x) to well-being: {income_contribution:.2f}")
print(f"Contribution of Education (y) to well-being: {education_contribution:.2f}")
print(f"Contribution of Crime Rate (z) to well-being: {crime_rate_contribution:.2f}")
print(f"Contribution of Employment (w) to well-being: {employment_contribution:.2f}")
print(f"Contribution of Housing (v) to well-being: {housing_contribution:.2f}")
print(f"Contribution of Healthcare (u) to well-being: {healthcare_contribution:.2f}")

# Step 8: Console Output (Copy/Paste)
Results:
Contribution of Income (x) to well-being: 9.74
Contribution of Education (y) to well-being: 8.37
Contribution of Crime Rate (z) to well-being: 14.14
Contribution of Employment (w) to well-being: 31.81
Contribution of Housing (v) to well-being: 4.93
Contribution of Healthcare (u) to well-being: -1.23
```