# Task

Create a prototype pricing model that can go through further validation and testing before being put into production

# Code

# A simple program to calculate gas storage contract value

# Import the date function from datetime module

from datetime import date

# Function to calculate the contract value

def calculate\_contract\_value(inject\_dates, inject\_prices, extract\_dates, extract\_prices, daily\_rate, storage\_cost, max\_volume):

stored\_gas = 0

total\_cost = 0

total\_revenue = 0

# Calculate injection costs and update stored gas

for i in range(len(inject\_dates)):

if stored\_gas + daily\_rate <= max\_volume:

stored\_gas += daily\_rate

total\_cost += daily\_rate \* inject\_prices[i]

print(f"Injected gas on {inject\_dates[i]} at price ${inject\_prices[i]}")

else:

print(f"Couldn't inject gas on {inject\_dates[i]} - storage full")

# Calculate extraction revenue and update stored gas

for i in range(len(extract\_dates)):

if stored\_gas >= daily\_rate:

stored\_gas -= daily\_rate

total\_revenue += daily\_rate \* extract\_prices[i]

print(f"Extracted gas on {extract\_dates[i]} at price ${extract\_prices[i]}")

else:

print(f"Couldn't extract gas on {extract\_dates[i]} - not enough gas")

# Calculate total days of storage

storage\_days = (max(max(inject\_dates), max(extract\_dates)) - min(min(inject\_dates), min(extract\_dates))).days

storage\_months = storage\_days // 30 # Approximate number of months

total\_storage\_cost = storage\_months \* storage\_cost

# Calculate and return the final value

contract\_value = total\_revenue - total\_cost - total\_storage\_cost

return contract\_value

# Example usage

inject\_dates = [date(2022, 1, 1), date(2022, 2, 1), date(2022, 3, 1)]

inject\_prices = [20, 22, 21]

extract\_dates = [date(2022, 4, 1), date(2022, 5, 1), date(2022, 6, 1)]

extract\_prices = [25, 23, 26]

daily\_rate = 1000 # cubic feet per day

storage\_cost = 5000 # dollars per month

max\_volume = 5000 # maximum storage in cubic feet

result = calculate\_contract\_value(inject\_dates, inject\_prices, extract\_dates, extract\_prices, daily\_rate, storage\_cost, max\_volume)

print(f"\nThe value of the contract is: ${result}")