Math 10 - Cumulative Project I - Introduction To Algorithms

This project will focus on the competencies *problem solving*, *technology*, and *communication*. The goal of the project is designing algorithms/programs/general solutions that could be used to complete each task. Some tasks are significantly more challenging then others.

For more information on python, check out the website https://www.python.org

---

Tasks - Measurement

Volume Computation (Required)

An engineer is designing a pool. If the pool has a length l, width w, and depth d.

i. Write a general solution for volume v in terms of l, w, and d.
Solution: $v = l \cdot w \cdot d$

ii. Write a python program that takes the dimensions of pool in **feet** and will calculate the volume of the pool in m^3, $feet^3$, and $inches^3$.
Possible Solution:

```
print("In meters, this is a program that will determine the volume of a pool given length, width, and depth")

l = int(input("Input A Length: "))
w = int(input("Input A Width: "))
d = int(input("Input A Depth: "))

print("The Volume Is: ")
print(l*w*d)
quit()
```

Lighthouse Problem

A lighthouse is being built so that is can spread its light over an area of a m^2, where $a \in \mathbb{Q}$ and $a > 0$. What should the engineers of the lighthouse make the height h in **meters** if the light is to reach/cover an area of a m^2, and the maximum distance a beam of light can travel is l km, where $l \in \mathbb{Q}$ and $l > 0$?

i. Write a general solution for h, in terms of a, and l.
Solution: $h = \sqrt{\left(\frac{l}{1000}\right)^2 - \left(\frac{a}{\pi}\right)}$

ii. Write a python program that will calculate h, given a, and l.
Hint: You will need to import `math` in python, in order to use math.pi for π
Possible Solution:

```
import math
print("This Is A Program That Determines The Height Of A Lighthouse, Given The Area It Is To Cover:")

# Input Values that will be known
```

```
a = int(input("Input An Area: "))
l = int(input("Input The Maximum Distance A Beam Of Light Can Travel In Km: "))
l = l/1000 # Convert Kilometers To Meters

# Create Variable for Pi From Math
pi = math.pi

#Determine radius of circular trajectory
r = (a/pi)**(1/2)

# Using the radius, and maximum beam length determine the height
h = (l**2-r**2)**(1/2)

print("The Height Of The Tower Will Be", h, "meters.")


quit()
```

Savings Problem

You are saving money to purchase an item. The item costs a dollars, where $a \in \mathbb{Q}$, $a > 0$. In your bank account you have s dollars, where $s \in \mathbb{Q}$, $s > 0$. At your current job you are making m dollars a month, where $m \in \mathbb{Q}$, $m > 0$.

i. Write a general solution for l, the length in **years** it will take before you are able to purchase the item. Write your solution in terms of a, s, and m.
Solution: $l = \frac{a-s}{12m}$

ii. Write a python program that will take the cost of the item a, the amount of money you have saved s, and your monthly income m, and give the length of time until you are able to purchase the item.
Possible Solution:

```
print("This Is A Program That Determines How Long It Will Take To Save Enough Money to Make
    Purchase, Given Savings and Monthly Income. :")

# Input Values that will be known
a = int(input("Input How Much Purchase Costs: "))
s = int(input("Input Ammount of Money You Have Saved: "))
n = int(input("Input Monthly Salary:"))

# Calculate The Time In Months:
t_months = (a-s)/n

# Convert Months To Years:
t_years = t_months/12

print("The Time It Will Take To Save For The Purchase Will Be", t_years, "years.")


quit()
```

Converting Between Grams and Moles

In Science class, you have been determining the molar mass of different molecular compounds. Given m grams of a particular compound, $m \in \mathbb{Q}$, $m > 0$, can you convert weight into n moles, $n \in \mathbb{Q}$, $n > 0$?

i. Write an algorithm/general solution that will take a molar mass of a compound (M) and a weight in **kilograms** (m_{kg}), and convert the weight to moles (n).
Solution: $m = \frac{1000g}{M}$

ii. Write a python program that will take a molar mass of a compound (M) and a weight in grams (g), and convert the weight to moles (m).
Possible Solution:

```
print("This Is A Program That Determines A Weight In Moles Given A Molar Mass Of Compound, And A
    Weight In Kilograms:")

# Input Values that will be known
M = int(input("Input Molar Mass Of Compound: "))
```

```
kg = int(input("Input Weight In Kilograms: "))

# Convert kg to g:
g = kg*1000

# Find Weight In Moles:
n = m/M

print("The Weight Will Be", m, "Moles")


quit()
```

Kinetic Energy Of A Moving Object

The Kinetic Energy of a moving object is given by the equation: $E_k = \frac{1}{2}mv^2$. Energy using the unit *joules*. When defining a joule using mass and velocity:

$$1\ joule\ = 1kg * \left(\frac{m}{s}\right)^2$$

i. Write a general solution for v in **km/h** if you are given the mass of an object m in kg, and its kinetic energy E_k in *joules*.
Solution: $v = 3.6\sqrt{\frac{2E_k}{m}}$

ii. Write a python program that will find the velocity of a moving object in **km/h** if you are given the mass of an object m and its kinetic energy E_k in *joules*.
Possible Solution:

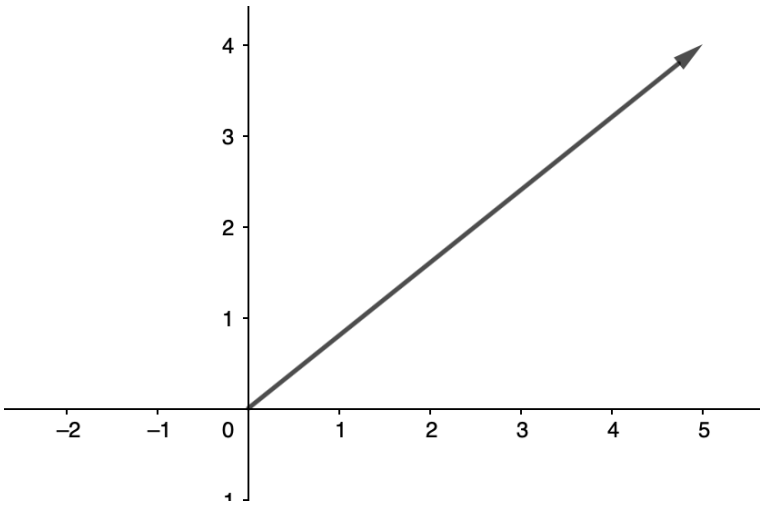
```
print("This Is A Program That Determines The Velocity Of A Moving Object Given Its Mass In Kg,
    and Its Kinetic Energy In Joules:")

# Input Values that will be known
Ek = int(input("Input Kinetic Energy Of Object: "))
m = int(input("Input Weight Of Object In Kilograms: "))

# Compute v
vms = (2*Ek/m)**(1/2) # velocity in m/s
v = 3.6*vms # velocity in km/h
print("The Velocity Of The Moving Object Will Be", v, "km/h")

quit()
```

Pythagorean Theorem In 3-Dimensions

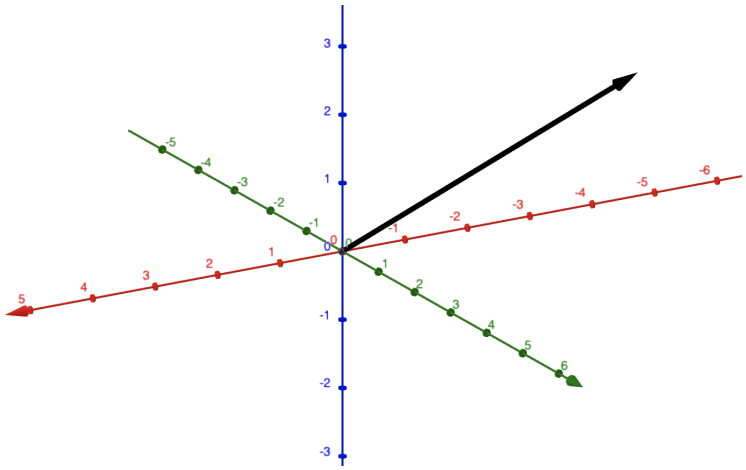
In your mathematics career, you have learned the **Pythagorean Theorem**:

$$a^2 + b^2 = c^2$$

This is often used to represent the hypotenuse of a right triangle, given the opposite and adjacent side. In other words, it represents a length of an arrow, given an x-coordinate, and a y-coordinate:



In this task, we would like to this formula to 3-dimensions. In other words, to find the length of an arrow in 3D, or $\mathbb{R}^3$:



i. Write a general formula for the length of the arrow, given its x, y, and z coordinates:
Solution: $l = \sqrt{x^2 + y^2 + z^2}$

ii. Write a python program that will compute the length of an arrow in 3-dimensions, given its x, y, and z coordinates.

```
print("This Is A Program That Determines The Length Of A Vector/Arrow In 3D Given An X, Y, and Z
    Coordinate:")

# Input Values that will be known
xcoord = int(input("Input An X-Coordinate: "))
ycoord = int(input("Input A Y-Coordinate: "))
zcoord = int(input("Input A Z-Coordinate: "))

# Compute l
l = (xcoord**2+ycoord**2+zcoord**2)**(1/2)
print("The Length Of The Vector Will Be", l)

quit()
```

iii. **Challenge:** Extend your formula to find the magnitude of an arrow/vector in $\mathbb{R}^n$.
Solution: $l = \sqrt{\sum_{i=1}^{n} x_i^2}$.

Tasks - Algebra And Number

Planetary Alignment Problem For Unknown Galaxy

In a solar system there are n planets, where $n \in \mathbb{N}$. Each planet has a unique orbital period T_i, $i = 1, 2, 3, \ldots, n$. If the planets are orbitally aligned at time t_o, design an algorithm to determine the time t when planets will align again. A sample solution is shown below:

i. Create a general algorithm that can be used to solve this problem for n different planets.
Solution: Find the lowest common multiple between the periods of the planets.

ii. Write a python script that will determine the next orbital alignment in a galaxy that has $n = 2$ planets.
Hint: You might want to use Python's gcd() function, that finds the greatest common factor of two numbers.
Sample Solution:

```
import math
print("This Is A Program That Will Determine The Time Of The Next Planetary Allignment Of 2
    Planets, Given Their Orbital Periods In Years")

# Input Values that will be known
T1 = int(input("Input the Orbital Perid For The First Planet: "))
T2 = int(input("Input the Orbital Perid For The Second Planet: "))

# Function That Finds LCM
def lcm(a,b): # Define Function
    if a%b == 0:
        lowestcm = a
        return lowestcm
    elif b%a == 0:
        lowestcm = b
        return lowestcm
    else :
        lowestcm = a*b/(math.gcd(a,b)) # Find LCM
        return lowestcm

print("The Orbital Alignment Will Occur In", lcm(T1,T2), "Years")
quit()
```

iii. **Challenge I:** Write a python function, instead of using gcd.

iv. **Challenge II:** Write a python script that will determine the next orbital alignment in a galaxy that has n planets.

The Locker Problem

In a school, n students are assigned n lockers. Students are assigned a number i, $i = 1, 2, 3, \ldots, n$, based on their locker number. The student assigned locker 1 opens all n lockers, the student assigned the number 2 closes all lockers that are multiples of 2. The i^{th} students opens closed lockers and closes open lockers that are multiples of i. What lockers are open after the i^{th} student goes through the lockers?

i. What lockers will remain open after the n^{th} student passes through?
Solution: Lockers that are perfect squares will remain open.

ii. Write a python script that will take the number of students n, and output the lockers that will remain open after the n^{th} student has passed through.
Sample Solutions: The first solutions assumes that students know the lockers that will open:

```
print("This Is A Program That Will Determine The Time Of The Next Planetary Allignment Of 2
    Planets, Given Their Orbital Periods In Years")

# Input Values that will be known
print("This Is A Program That Determines The Lockers That Will Remain Open After The Last
    Student Has Passed Through:")

# Input Values that will be known
n = int(input("Enter The Number Of Students/Lockers: "))
```

```
8
9  print("Lockers That Will Be Open Are:")
10 for i in range(1,n+1):
11     if (i**(1/2))%1 == 0 :
12         print(i)
13
14 quit()
```

The second solution is if the students want to iteratively solve the problem. This is not recommended as it is **much harder**:

```
1  print("This Is A Program That Will Determine Which Lockers Will Be Open After All Students Have
       Gone Through:")
2
3  # Input Values that will be known
4  n = int(input("Input The Number Of Students: "))
5
6  L={} # Make Empty Dictionary, Dictionaries Have Keys and Values
7  print("The Lockers That Are Open Are: ")
8  for i in range(1, n+1): # For Every Locker Number
9      L[i] = False
10     for j in range (1,n+1): # For Every Locker
11         if i%j == 0: # Does each locker divide evenly by i?
12             L[i]= not L[i] # Open/Close Door For Each Locker
13     if L[i]: # If Locker Is Open (True) Then Print
14         print(i)
15
16 print(L) # To See How It Works
17
18 quit()
```

Iterative Approximation To A Radical

Given n, where $n \in \mathbb{W}$, we can approximate $\sqrt{2}$ using what is known as the *Babylonian Method.* First research the algorithm and write a brief explanation about how it works. Next, write a program that will carry out the algorithm, and approximate $\sqrt{2}$ with precision. A sample solution is shown below:

The Babylonian Method is used to approximate $\sqrt{a}$. The Algorithm uses the series:

$$x_{n+1} = \frac{x_o + \frac{S}{x_o}}{2}$$

For example, If I were to approximate $\sqrt{5}$:

i. Begin by guessing at what the number might be. We know that $2 < \sqrt{5} < 3$, so we will guess 2.5

ii. Next, we apply the Babylonian algorithm:

$$\begin{aligned} x_1 &= \frac{2.5 + \frac{5}{2.5}}{2} = 2.25 \\ x_2 &= \frac{2.25 + \frac{5}{2.25}}{2} = 2.236111 \\ x_3 &= \frac{2.236111 + \frac{5}{2.236111}}{2} = 2.236068 \\ x_4 &= \frac{2.236068 + \frac{5}{2.236068}}{2} = 2.236068 \end{aligned}$$

So $\sqrt{5} \approx 2.236068$.

i. Write a program that will approximate $\sqrt{a}$ using the Babylonian method.
Possible Solution:

```
print("This Is A Program That Will Approximation to Sqrt(a) to 10 decimal Places Using the
    Babalonian method:")

# Input Values that will be known
a = int(input("Input a: ")) # input a
guess = int(input("Input an inital guess: ")) # input initial guess
guess1 = guess+1 # pick any number larger than guess

while abs(guess-guess1)>0.00000000000000000000000000000000000000000000000000000000000000000000000000000001 :
    guess1 = guess;
    guess = (guess + (a/guess))/2;
    print(guess);

print("The approximation is: ")
print(guess)

quit()
```

ii. Research alternative algorithms that are used online, and write a brief description of another.
Solution: Answers May Vary

What Is The Probability Of Winning?

When playing a certain game a person has probability p, $0 \leq p \leq 1$ of winning each round.

i. What is the probability of losing each round in the game?
Solution: 1-P(Winning) = 1 - p

ii. If the person plays the game for n rounds, what is the probability that they will only win on the n^{th} round? Write a general solution in terms of n and p.
Solution: Geometric probability distribution - $P(\text{Win On Last Round}) = (1-p)^{n-1}p$

iii. Write a program that takes the probability of winning a game, and determines the probability of winning the game only on the n^{th} round.
Possible Solution:

```
print("This Is A Program That Determines The Probability Of Winning The Game On The Last Round,
    Given n Rounds And Probability Of Winning p")

# Input Values that will be known
n = int(input("Enter The Number Of Rounds Played: "))
p = (input("Enter The Probability Of Winning Each Round: "))
p = float(p)

# Find Probability Of Losing
q = 1-p

# Find Probability Of Winning only on the last round
probability = p*(q**(n-1))

print("The probability of winning will be", probability)

quit()
```

iv. **Challenge:** If the person plays the game for n rounds, determine the probability that they will win once regardless of round. Write a general solution in terms of n and p.
Solution: Binomial Probability Distribution - $\binom{n}{1}p(1-p)^{n-1}$

Fission In A Nuclear Reactor

For a brief history on nuclear energy: https://www.youtube.com/watch?v=rcOFV4y5z8c

In a nuclear fission reaction, an atom is split by an accelerated neutron. The splitting of the atom causes a tremendous output of energy. Each atom that is split produces n new neutrons that will split n new atoms. Every time a group of atoms is split, it is called a generation g.

i. In a nuclear fission reaction, write the general formula for the number of fissions that occurs in a nuclear reaction, given that each fission produces n neutrons, and there have been g generations.
Solution: $n_{fissions} = n^g$

ii. Write a python program that will take the number of neutrons produced by each fission n, the amount of generations, and outputs the number of fissions.
Possible Solution:

```
print("This Is A Program That Determines The Number Of Fissions After g Generations, Given Each
    Fission Produces n Neutrons")

# Input Values that will be known
n = int(input("Enter The Number Of Neutrons Produced At Each Fission: "))
g = int(input("Enter The Number Generations That Have Occured: "))

# Compute the Number Of Fissions
f = n**g

print("The number of fissions will be ", f)

quit()
```

Magical Trap In Harry Potter

In *Harry Potter And The Deathly Hallows Part II*, there is a scene where Ron, Harry, and Hermione have broken into the magical bank Gringotts in search of a Horcrux: https://www.youtube.com/watch?v=mcMjdSk9EfY. The Gemino curse has been applied to the various treasure in the room. After touching a cup, it will multiply to become n cups.

i. A cup occupies v_{cup} m^3 of a room, and after being touched spontaneously multiplies to be n cups every division. Write a general solution for how much volume will be occupied after d divsions.
Solution: $v_{occupied} = v_{cup}n^d$

ii. Write a python program that will take the multiplication factor n, the number of divisions $t = d$, and will output the volume that will be occupied by all the cups.

```
print("This Is A Program That Determines The Volume That Will Be Occupied By the Cups After d
    Divisions, Given The Division Factor and Volume Of Origional Cup:")

# Input Values that will be known
n = int(input("Enter The Division Factor Of The Curse "))
d = int(input("Enter the Number Of Divsions "))
vcup = int(input("Enter The Volume Of The Cup: "))

# Compute Volume
v = vcup*(n**d)

print("The Volume Will Be", v)

quit()
```

iii. **Challenge:** If the room has a total volume of v_{room} m^3, how many divisions will it take until the room is entirely full of cups? Write a general solutions for d in terms of v_{cup}, v_{room}, and n
Solution: $d = \log_n\left(\frac{v_{room}}{v_{cup}}\right)$