Finance Question

- **Write a 500-word explanation of Bitcoin stock-to-flow model and make an argument for why it is a bad model?**

The stock to flow model was proposed by Plan B which compares Bitcoin to scare commodities like gold, silver or platinum because they retain their value for a long period of time. Plan B explained the difference between gold and bitcoin with respect to other consumable commodities like zinc, copper, nickel etc and identifies that the high stock to flow ratio of gold and bitcoin differentiates them from other consumable commodities which also relates to the fact that the process of searching and mining gold and bitcoin is quite expensive and takes time. Stock to flow is defined as a relationship between the annual production and current stock that is out there. It is used to evaluate the current stock of a commodity (i.e the total amount available that moment) against the flow of new production (i.e amount mined that year). A summary of the stock to ratio of Bitcoin as proposed by Plan B is that the price of Bitcoin would experience an exponential increase since it follows the stock to flow ratio line as observed from previous data using linear regression.

One of the major point backing this model is the fact that it relies solely on the scarcity of supply, from basic economics, the law of demand and supply interact to determine the actual market prices and volume of goods that are traded on a market, not just the law of supply can validate or predict the price of Bitcoin, Since demand wasn't put into consideration in determining the price, the model is not genuine.

Plan B compares Bitcoin with Gold, Bitcoin has just been in the market for about 10 years or less while Gold has been in the market for over 115 years. A model is considered very much valid depending on how much data has been analysed, so comparing the analysis of a 10 year old data with that of a 115 year old data can never result in the same output.

Another limitation of this model is that it doesn't consider some economic unforeseen contingencies. Theoretically, historical data cannot account for some unknown events that might arise later.

Research records that mining bitcoin consumes a lot of energy and computing power, obviously an increase in price would lead to an increase in energy consumption. When the rate of energy consumed begins to pose a threat to the environment, many entrenched powers: regulators, governments, banks, tax authorities, Western Union, financial services, environmentalists, gold bugs, the FBI would try and reduce the adoption of bitcoin by placing a temporary ban or tax on it, this would lead to a decrease in price if this happens.

This model also claims that Bitcoin's volatility should also decrease over time but doesn't give a concrete fact to back this up. From history, Bitcoin is known for its notorious price changes thus making it highly volatile. On a macro level, there might be a decreasing volatility but since Bitcoin has been priced in a free market from inception, it means the price is mostly self regulated making users, traders and speculators a major determining factor of its price, therefore increasing its volatility.

- **(Please show your workings). Yara Inc is listed on the NYSE with a stock price of $40 - the company is not known to pay dividends. We need to price a call option with a strike of $45 maturing in 4 months. The continuously-compounded risk-free rate is 3%/year, the mean return on the stock is 7%/year, and the standard deviation of the stock return is 40%/year. What is the Black-Scholes call price?**

Solution

Black-Scholes Call price (C) = $SN(X_1)$ - $\beta N(X_2)$

S = Stock Price = $40

E = exercise price = $45

β = Bond Price = Ee^-(r_fT) = 45*e^(-0.03*1/3) = 44.5523

σ = standard deviation = 40%

T = 4/12 = 1/3

r_f= risk free rate = 3%

$$X_1 = \frac{In(S/)}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}$$

$$X_2 = \frac{In(S/)}{\sigma\sqrt{T}} + \frac{1}{2}\sigma\sqrt{T}$$

$$X_1 = \frac{In(40/44.5523)}{0.4\sqrt{1/3}} + \frac{1}{2} * 0.4\sqrt{1/3}$$

X_1 = -0.3513

$N(X_1)$ = probability that a value less than "*X*" will occur in a standard normal distribution = 0.3627

$$X_2 = \frac{In(40/44.5523)}{0.4\sqrt{1/3}} - \frac{1}{2} * 0.4\sqrt{1/3}$$

$X_2 = -0.5822$

$N(X_2)$ = probability that the call will be exercised = 0.2802

C = $SN(X_1)$ - $\beta N(X_2)$

C = (40*0.3627) - (44.5523*0.2802)

C = 2.0244

Computer Science Question

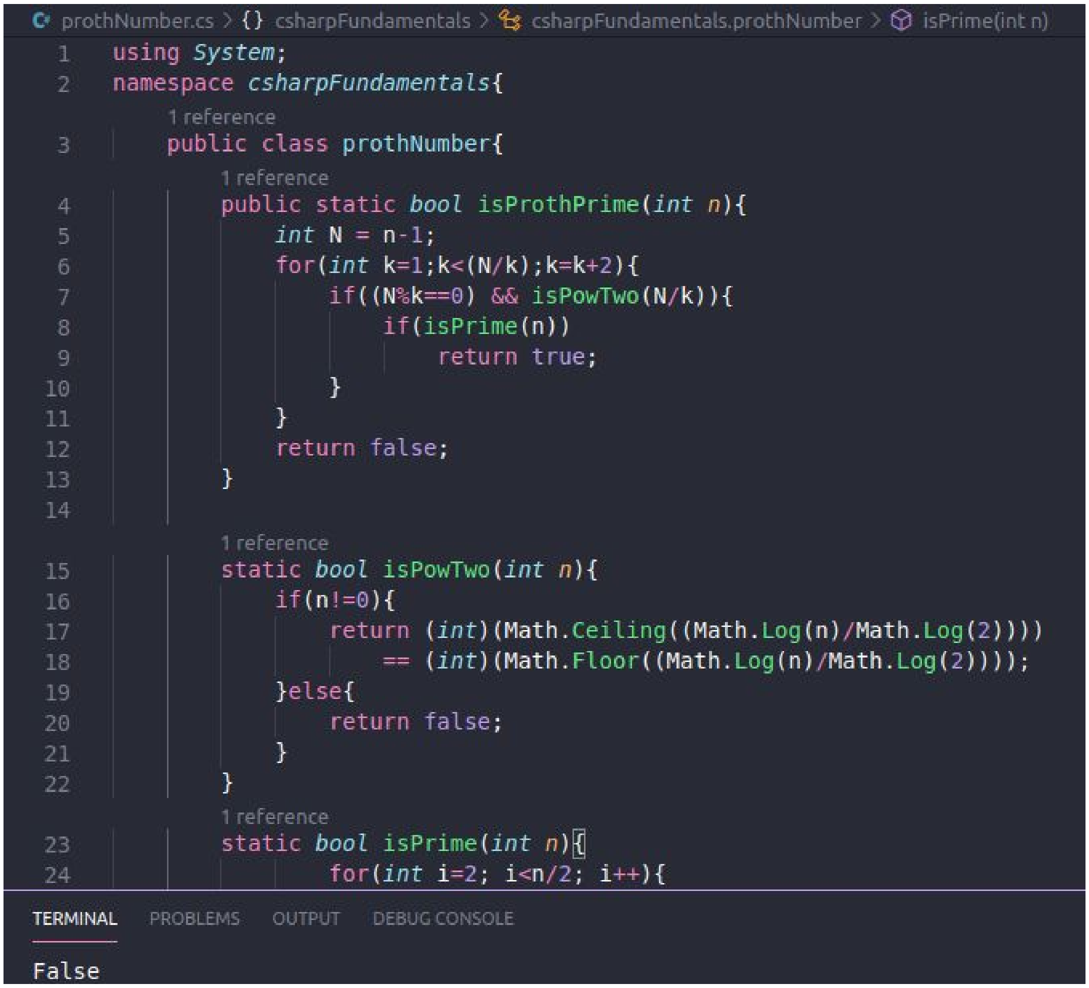
- Why is it a bad idea to use a recursion method to find the fibonacci of a number?

A recursive method is when a method calls itself within its own definition. It keeps calling itself over until a base condition is met to break the loop. Using recursive methods to find fibonacci is a bad idea because of the following reasons:

1. It increases time complexity: In the case of fibonacci,the same value is calculated over and over again, thus increasing the time to complete the task.
2. Whenever a recursive call is made, a stack space is allocated to store the variables, if a very large fibonacci series is to be calculated, more memory is consumed during such computation which may lead to memory or stack overflow.
3. An increase in computation time makes it slow for very large fibonacci series.

- Write a function that takes in a Proth Number and uses Proth's theorem to determine if said number is prime? You can write this in any programming language but C/C++/Golang are preferred.

Code is attached with the document.


prothNumber.cs > {} csharpFundamentals > csharpFundamentals.prothNumber > isPrime(int n)
1 reference
TERMINAL
PROBLEMS
OUTPUT
DEBUG CONSOLE

```
using System;
namespace csharpFundamentals{
    public class prothNumber{
        public static bool isProthPrime(int n){
            int N = n-1;
            for(int k=1;k<(N/k);k=k+2){
                if((N%k==0) && isPowTwo(N/k)){
                    if(isPrime(n))
                        return true;
                }
            }
            return false;
        }

        static bool isPowTwo(int n){
            if(n!=0){
                return (int)(Math.Ceiling((Math.Log(n)/Math.Log(2))))
                    == (int)(Math.Floor((Math.Log(n)/Math.Log(2))));
            }else{
                return false;
            }
        }
        static bool isPrime(int n){
                for(int i=2; i<n/2; i++){
```

```
False
```

Maths

- (Please show your workings). Over all real numbers, find the minimum value of a positive real number, y such that y=sqrt((x+6)2+25)+sqrt((x−6)2+121)

Solution

y = $\sqrt{(x+6)^z+25}+\sqrt{(x-6)^2+121}$

If y is a positive real no, that is y > 0

Therefore,

$\sqrt{(x+6)^z+25}+\sqrt{(x-6)^2+121}$ > 0

$\sqrt{(x+6)^z+25} > -(\sqrt{(x-6)^2+121})$

=> $\left((x+6)^2+25\right)^{\frac{1}{2}} > -\left((x-6)^2+121\right)^{\frac{1}{2}}$

Squaring both sides

$\left((x+6)^2+25\right) > -\left((x-6)^2+121\right)$

$(x^2+12x+36+25) > -(x^2-12x+36+121)$

$(x^2+12x+61) > -(x^2-12x+157)$

$(x^2+12x+61) > -x^2+12x-157$

$2x^2+0x+218$ > 0 $\equiv y=2x^2+0x+218$

expressing as a quadratic equation in th form

$y = ax^2+bx+c$

The value of $\frac{-b}{2a}$ tells the x value of the vertex of the curve which is also the minimum value

So x = $\frac{-b}{2a}$ = $\frac{-0}{2(2)}$ = 0

Therefore the minimum value = 0

Substitute the value of x in equation y above

y = $\sqrt{(0+6)^z+25}+\sqrt{(0-6)^2+121}$

y = $\sqrt{36+25}+\sqrt{36+121}$

y = 7.8 + 12.5

y = 20.8

So, the minimum value of a positive real number y = 20.8 is x = 0