Perfect Plan B

Learn, grow and become leaders of tomorrow

Problem 1 : Python Program to add two numbers

Example:

Input: num1 = 5, num2 = 3
Output: 8

Input: num1 = 13, num2 = 6
Output: 19

Solution 1 : Python Program to add two numbers

```
# Python3 program to add two numbers

num1 = 15
num2 = 12

# Adding two nos
sum = num1 + num2

# printing values
print("Sum of {0} and {1} is {2}" .format(num1, num2, sum))
```

Solution 2 : Python Program to add two numbers

```
# Python3 program to add two numbers

number1 = input("First number: ")
number2 = input("\nSecond number: ")

# Adding two numbers
# User might also enter float numbers
sum = float(number1) + float(number2)

# Display the sum
# will print value in float
print("The sum of {0} and {1} is {2}" .format(number1, number2, sum))
```

Problem 2 : Python Program for factorial of a number

$$n! = n * (n-1) * (n-2) * \ldots\ldots * 1$$

Examples :

$$4! = 4*3*2*1 = 24$$

$$6! = 6*5*4*3*2*1 = 720$$

Solution 1 : Python Program for factorial of a number (Iterative)

```
# Python 3 program to find
# factorial of given number
def factorial(n):
    if n < 0:
        return 0
    elif n == 0 or n == 1:
        return 1
    else:
        fact = 1
        while(n > 1):
            fact *= n
            n -= 1
        return fact

# Driver Code
num = 5;
print("Factorial of",num,"is",
factorial(num))
```

Solution 2 : Python Program for factorial of a number (Recursive)

```
# Python 3 program to find
# factorial of given number
def factorial(n):

    # single line to find factorial
    return 1 if (n==1 or n==0) else n * factorial(n - 1);

# Driver Code
num = 5;
print("Factorial of",num,"is",
factorial(num))
```

Solution 3 : Python Program for factorial of a number

```
# Python 3 program to find
# factorial of given number

def factorial(n):

    # single line to find factorial
    return 1 if (n==1 or n==0) else n * factorial(n - 1)


# Driver Code
num = 5
print ("Factorial of",num,"is",
    factorial(num))
```

Problem 3 : Python Program for simple interest

Simple interest formula is given by:

Simple Interest = (P x T x R)/100

Where,

P is the principle amount

T is the time and

R is the rate

```
EXAMPLE1:
Input : P = 10000
        R = 5
        T = 5
Output :2500
We need to find simple interest on
Rs. 10,000 at the rate of 5% for 5
units of time.

EXAMPLE2:
Input : P = 3000
        R = 7
        T = 1
 Output :210
```

Solution : Python Program for simple interest

```
# Python3 program to find simple interest
# for given principal amount, time and
# rate of interest.


def simple_interest(p,t,r):
        print('The principal is', p)
        print('The time period is', t)
        print('The rate of interest is',r)

        si = (p * t * r)/100

        print('The Simple Interest is', si)
        return si

# Driver code
simple_interest(8, 6, 8)
```

Problem 4 : Python Program for compound interest

Formula to calculate compound interest annually is given by:

Compound Interest = P(1 + R/100) t

Where,

P is principle amount

R is the rate and

T is the time span

```
Input : Principle (amount): 1200
        Time: 2
        Rate: 5.4
Output : Compound Interest =
1333.099243
```

Solution : Python Program for compound interest

```
# Python3 program to find compound
# interest for given values.

def compound_interest(principle, rate, time):

    # Calculates compound interest
    CI = principle * (pow((1 + rate / 100), time))
    print("Compound interest is", CI)

# Driver Code
compound_interest(10000, 10.25, 5)
```

Problem 5 : Python Program to check Armstrong Number

```
Input : 153
Output : Yes
153 is an Armstrong number.
1*1*1 + 5*5*5 + 3*3*3 = 153

Input : 120
Output : No
120 is not a Armstrong number.
1*1*1 + 2*2*2 + 0*0*0 = 9

Input : 1253
Output : No
1253 is not a Armstrong Number
1*1*1*1 + 2*2*2*2 + 5*5*5*5 + 3*3*3*3 =
723

Input : 1634
Output : Yes
 1*1*1*1 + 6*6*6*6 + 3*3*3*3 + 4*4*4*4 =

 1634
```

Solution : Python Program to check Armstrong Number

```
# Python program to determine whether the
number is
# Armstrong number or not

# Function to calculate x raised to the power y
def power(x, y):
    if y==0:
        return 1
    if y%2==0:
        return power(x, y/2)*power(x, y/2)
    return x*power(x, y/2)*power(x, y/2)

# Function to calculate order of the number
def order(x):

    # variable to store of the number
    n = 0
    while (x!=0):
        n = n+1
        x = x/10
    return n
```

```
# Function to check whether the given
number is
# Armstrong number or not
def isArmstrong (x):
    n = order(x)
    temp = x
    sum1 = 0
    while (temp!=0):
        r = temp%10
        sum1 = sum1 + power(r, n)
        temp = temp/10

    # If condition satisfies
    return (sum1 == x)


# Driver Program
x = 153
print(isArmstrong(x))
x = 1253
print(isArmstrong(x))
```

Problem 6 : Python Program for Program to find area of a circle

```
Area = pi * r2
 where r is radius of circle
```

Solution : Python Program for Program to find area of a circle

```
# Python program to find Area of a circle

def findArea(r):
        PI = 3.142
        return PI * (r*r);

# Driver method
print("Area is %.6f" % findArea(5));
```

Problem 7 : Python program to print all Prime numbers in an Interval

Given two positive integer start and end. The task is to write a Python program to print all Prime numbers in an Interval.

Definition: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first few prime numbers are {2, 3, 5, 7, 11, ...}.

Solution : Python program to print all Prime numbers in an Interval

```
# Python program to print all
# prime number in an interval

start = 11
end = 25

for val in range(start, end + 1):
        if val > 1:
                for n in range(2, val//2 + 2):
                        if (val % n) == 0:
                                break
                        else:
                                if n == val//2 + 1:
                                        print(val)
```

Problem 8 : Python program to check whether a number is Prime or not

Definition: A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself. The first few prime numbers are {2, 3, 5, 7, 11, ...}.


```
Input:  n = 11

Output: true

Input:  n = 15

Output: false
```

Solution : Python program to check whether a number is Prime or not

```
# Python program to check if
# given number is prime or not

num = 11

# If given number is greater than 1
if num > 1:

# Iterate from 2 to n / 2
for i in range(2, num//2):

        # If num is divisible by any number between
        # 2 and n / 2, it is not prime
        if (num % i) == 0:
                print(num, "is not a prime number")
                break
else:
        print(num, "is a prime number")

else:
print(num, "is not a prime number")
```

Solution : Python program to check whether a number is Prime or not

```
# A optimized school method based
# Python3 program to check
# if a number is prime


def isPrime(n) :

        # Corner cases
        if (n <= 1) :
                return False
        if (n <= 3) :
                return True

        # This is checked so that we can
skip
        # middle five numbers in below
loop
        if (n % 2 == 0 or n % 3 == 0) :
                return False
        i = 5
        while(i * i <= n) :
                if (n % i == 0 or n % (i + 2) == 0) :
                        return False
                i = i + 6

        return True


# Driver Program
if (isPrime(11)) :
        print(" true")
else :
        print(" false")

if(isPrime(15)) :
        print(" true")
else :
        print(" false")
```

Problem 9 : Python Program for n-th Fibonacci number

In mathematical terms, the sequence Fn of Fibonacci numbers is

defined by the recurrence relation

```
Fn = Fn-1 + Fn-2
```

with seed values

```
F0 = 0 and F1 = 1.
```

Hint : Recursion

Solution : Python Program for n-th Fibonacci number

```
# Function for nth Fibonacci number

def Fibonacci(n):
        if n<0:
                print("Incorrect input")
        # First Fibonacci number is 0
        elif n==1:
                return 0
        # Second Fibonacci number is 1
        elif n==2:
                return 1
        else:
                return Fibonacci(n-1)+Fibonacci(n-2)

# Driver Program

print(Fibonacci(9))
```

Solution : Python Program for n-th Fibonacci number

```
# Function for nth fibonacci number - Dynamic Programing
# Taking 1st two fibonacci nubers as 0 and 1

FibArray = [0,1]

def fibonacci(n):
        if n<0:
                print("Incorrect input")
        elif n<=len(FibArray):
                return FibArray[n-1]
        else:
                temp_fib = fibonacci(n-1)+fibonacci(n-2)
                FibArray.append(temp_fib)
                return temp_fib

# Driver Program

print(fibonacci(9))
```

Solution : Python Program for n-th Fibonacci number

```
# Function for nth fibonacci number - Space Optimisataion
# Taking 1st two fibonacci numbers as 0 and 1

def fibonacci(n):
        a = 0
        b = 1
        if n < 0:
                print("Incorrect input")
        elif n == 0:
                return a
        elif n == 1:
                return b
        else:
                for i in range(2,n):
                        c = a + b
                        a = b
                        b = c
                return b

# Driver Program

print(fibonacci(9))
```

Problem 10 : Python Program for printing Fibonacci numbers

In mathematical terms, the sequence Fn of Fibonacci numbers is

defined by the recurrence relation

```
Fn = Fn-1 + Fn-2
```


with seed values

```
F0 = 0 and F1 = 1.
```

Hint : Recursion

Solution : Python Program for printing Fibonacci numbers

```
# Function for nth Fibonacci number

def Fibonacci(n):
        if n<0:
                print("Incorrect input")
        # First Fibonacci number is 0
        elif n==1:
                return 0
        # Second Fibonacci number is 1
        elif n==2:
                return 1
        else:
                return Fibonacci(n-1)+Fibonacci(n-2)

# Driver Program

print(Fibonacci(9))
```

Solution : Python Program for printing Fibonacci numbers

```
# Function for nth fibonacci number - Dynamic Programing
# Taking 1st two fibonacci nubers as 0 and 1

FibArray = [0,1]

def fibonacci(n):
        if n<0:
                print("Incorrect input")
        elif n<=len(FibArray):
                return FibArray[n-1]
        else:
                temp_fib = fibonacci(n-1)+fibonacci(n-2)
                FibArray.append(temp_fib)
                return temp_fib

# Driver Program

print(fibonacci(9))
```

Solution : Python Program for printing Fibonacci numbers

```
# Function for nth fibonacci number - Space Optimisataion
# Taking 1st two fibonacci numbers as 0 and 1

def fibonacci(n):
        a = 0
        b = 1
        if n < 0:
                print("Incorrect input")
        elif n == 0:
                return a
        elif n == 1:
                return b
        else:
                for i in range(2,n):
                        c = a + b
                        a = b
                        b = c
                return b

# Driver Program

print(fibonacci(9))
```

Problem 11 : Python Program for How to check if a given number is Fibonacci number?

```
Input : 8

Output : Yes

Input : 34

Output : Yes

Input : 41

Output : No
```

Solution : Python Program for How to check if a given number is Fibonacci number?

```
# python program to check if x is a perfect square
import math

# A utility function that returns true if x is perfect square
def isPerfectSquare(x):
        s = int(math.sqrt(x))
        return s*s == x

# Returns true if n is a Fibinacci Number, else false
def isFibonacci(n):

        # n is Fibinacci if one of 5*n*n + 4 or 5*n*n - 4 or both
        # is a perferct square
        return isPerfectSquare(5*n*n + 4) or isPerfectSquare(5*n*n - 4)

# A utility function to test above functions
for i in range(1,11):
        if (isFibonacci(i) == True):
                print i,"is a Fibonacci Number"
        else:
                print i,"is a not Fibonacci Number "
```

Problem 12 : Program to print ASCII Value of a character

```
Input : a
```

```
Output : 97
```



```
Input : D
```

```
Output : 68
```

Solution : Program to print ASCII Value of a character

```
# Python program to print
# ASCII Value of Character

# In c we can assign different
# characters of which we want ASCII value

c = 'g'
# print the ASCII value of assigned character in c
print("The ASCII value of '" + c + "' is", ord(c))
```

Problem 13 : Python Program for Sum of squares of first n natural numbers

```
Input : N = 4

Output : 30

= 1 + 4 + 9 + 16

= 30
```



```
Input : N = 5

Output : 55
```

Solution : Python Program for Sum of squares of first n natural numbers

```
# Python3 Program to
# find sum of square
# of first n natural
# numbers


# Return the sum of
# square of first n
# natural numbers
def squaresum(n) :

        # Iterate i from 1
        # and n finding
        # square of i and
        # add to sum.
        sm = 0
        for i in range(1, n+1) :
                sm = sm + (i * i)

        return sm

# Driven Program
n = 4
print(squaresum(n))
```

Solution : Python Program for Sum of squares of first n natural numbers

```
# Python3 Program to
# find sum of square
# of first n natural
# numbers

# Return the sum of
# square of first n
# natural numbers
def squaresum(n) :
        return (n * (n + 1) * (2 * n + 1)) // 6

# Driven Program
n = 4
print(squaresum(n))
```

Solution : Python Program for Sum of squares of first n natural numbers

```
# Python Program to find sum of square of first
# n natural numbers. This program avoids
# overflow upto some extent for large value
# of n.y

def squaresum(n):
        return (n * (n + 1) / 2) * (2 * n + 1) / 3

# main()
n = 4
print(squaresum(n));
```

Problem 14 : Python Program for cube sum of first n natural numbers

```
Input : n = 5

Output : 225

13 + 23 + 33 + 43 + 53 = 225
```



```
Input : n = 7

Output : 784

13 + 23 + 33 + 43 + 53 +

63 + 73 = 784
```

Solution : Python Program for cube sum of first n natural numbers

```
# Simple Python program to find sum of series
# with cubes of first n natural numbers

# Returns the sum of series
def sumOfSeries(n):
        sum = 0
        for i in range(1, n+1):
                sum +=i*i*i

        return sum


# Driver Function
n = 5
print(sumOfSeries(n))
```

Solution : Python Program for cube sum of first n natural numbers

```
# A formula based Python program to find sum
# of series with cubes of first n natural
# numbers

# Returns the sum of series
def sumOfSeries(n):
        x = (n * (n + 1) / 2)
        return (int)(x * x)


# Driver Function
n = 5
print(sumOfSeries(n))
```

Solution : Python Program for cube sum of first n natural numbers

```
# Efficient Python program to find sum of cubes
# of first n natural numbers that avoids
# overflow if result is going to be within
# limits.

# Returns the sum of series
def sumOfSeries(n):
        x = 0
        if n % 2 == 0 :
                x = (n/2) * (n+1)
        else:
                x = ((n + 1) / 2) * n

        return (int)(x * x)


# Driver Function
n = 5
print(sumOfSeries(n))
```

Slack Invite Link

https://join.slack.com/t/perfect-plan-b/shared_invite/zt-drplefyv-x1vurrlFy98UOe1irCfLXw

Social Media Links

Facebook: https://www.facebook.com/IshanPlanB/

Twitter: https://twitter.com/PerfectPlanB1

Linkedin: https://www.linkedin.com/company/perfect-plan-b/

Instagram: https://www.instagram.com/perfect_plan_b/

Quora:
https://www.quora.com/q/hreieuophqgaswqv?ch=10&share=41d2481e&srid=ER3y0

Youtube: https://www.youtube.com/channel/UCQJFQlCdcq4XxJDqE3IqmbQ