Q1. Given an integer n, return *true* if it is a power of two. Otherwise, return *false*.

An integer n is a power of two, if there exists an integer x such that n == 2x.

class Solution:

def isPowerOfTwo(self, n: int) -> bool:

if(n==1):

return True

if(n<=0):

return False

elif(n%2==0):

return Solution().isPowerOfTwo(n/2)

else:

return False

Q2. Given a number n, find the sum of the first natural numbers.

def findSum(n):

if(n==1):

return 1

if(n<=0):

return 0

return n+findSum(n-1)

Q3. Given a positive integer, N. Find the factorial of N.

**def** factorial(n):

**if** n **==** 0:

**return** 1

**return** n **\*** factorial(n**-**1)

Q4. Given a number N and a power P, the task is to find the exponent of this number raised to the given power, i.e. N^P.

**def** power(N, P):

**if** P **==** 0:

**return** 1

**return** (N**\***power(N, P**-**1))

Q5. Find Maximum

**def** findMaxRec(A, n):

**if** (n **==** 1):

**return** A[0]

**return** max(A[n **-** 1], findMaxRec(A, n **-** 1))

Q6. Given first term (a), common difference (d) and a integer N of the Arithmetic Progression series, the task is to find Nth term of the series

**def** Nth\_of\_AP(a, d, N) :

**return** (a **+** (N **-** 1) **\*** d)

Q7. Permute all strings

**def** toString(List):

**return** ''.join(List)

**def** permute(a, l, r):

**if** l **==** r:

        print(toString(a))

**else**:

**for** i **in** range(l, r):

            a[l], a[i] **=** a[i], a[l]

            permute(a, l**+**1, r)

            a[l], a[i] **=** a[i], a[l]

Q8. Given an array, find a product of all array elements.

**def** multiply( array , n ):

    pro **=** 1

**for** i **in** range(n):

        pro **=** pro **\*** array[i]

**return** pro