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.

**Example 1:** Input: n = 1

Output: true

**Example 2:** Input: n = 16

Output: true

**Example 3:** Input: n = 3

Output: false

Solution :

def isPowerOfTwo(n):

if n <= 0:

return False

return (n & (n - 1)) == 0

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

**Example 1:**

Input: n = 3

Output: 6

**Example 2:**

Input : 5

Output : 15

Solution :

def sumOfFirstN(n):

return (n \* (n + 1)) // 2

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

**Example 1:**

Input: N = 5

Output: 120

**Example 2:**

Input: N = 4

Output: 24

Solution :

def factorial(N):

result = 1

for i in range(1, N + 1):

result \*= i

return result

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.

**Example 1 :**

Input: N = 5, P = 2

Output: 25

**Example 2 :** Input: N = 2, P = 5

Output: 32

Solution :

def factorial(n):

if n == 0 or n == 1:

return 1

else:

return n \* factorial(n - 1)

Given an array of integers **arr**, the task is to find maximum element of that array using recursion.

**Example 1:**

Input: arr = {1, 4, 3, -5, -4, 8, 6}; Output: 8

**Example 2:**

Input: arr = {1, 4, 45, 6, 10, -8}; Output: 45

Solution :

def findMax(arr, start, end):

if start == end:

return arr[start]

else:

mid = (start + end) // 2

max\_left = findMax(arr, start, mid)

max\_right = findMax(arr, mid + 1, end)

return max(max\_left, max\_right)

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.

**Example 1:**

Input : a = 2 d = 1 N = 5 Output : 6 The 5th term of the series is : 6

**Example 2:**

Input : a = 5 d = 2 N = 10 Output : 23 The 10th term of the series is : 23

Solution :

def findNthTerm(a, d, N):

nth\_term = a + (N - 1) \* d

return nth\_term

Given a string S, the task is to write a program to print all permutations of a given string.

**Example 1:**

***Input:***

S = “ABC”

***Output:***

“ABC”, “ACB”, “BAC”, “BCA”, “CBA”, “CAB”

**Example 2:**

***Input:***

S = “XY”

***Output:***

“XY”, “YX”

Solution :

def permute(index, chars, result):

if index == len(chars):

result.append(''.join(chars))

else:

for i in range(index, len(chars)):

chars[index], chars[i] = chars[i], chars[index]

permute(index + 1, chars, result)

chars[index], chars[i] = chars[i], chars[index] # backtrack

def generatePermutations(S):

chars = list(S)

result = []

permute(0, chars, result)

return result

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

**Example 1:**

Input : arr[] = {1, 2, 3, 4, 5} Output : 120 **Example 2:**

Input : arr[] = {1, 6, 3} Output : 18

Solution :

def productOfArray(arr):

product = 1

for num in arr:

product \*= num

return product