**Assignment 9 – Recursion**

**Question 1**

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

**Sol.**

def isPowerOfTwo(n):

if n <= 0:

return False

while n % 2 == 0:

n = n // 2

return n == 1

print(isPowerOfTwo(1)) # Output: True

print(isPowerOfTwo(16)) # Output: True

print(isPowerOfTwo(3)) # Output: False

**Question 2**

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

**Sol:**

def sumOfNaturalNumbers(n):

sum = (n \* (n + 1)) // 2

return sum

print(sumOfNaturalNumbers(3)) # Output: 6

print(sumOfNaturalNumbers(5)) # Output: 15

**Question 3**

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

**Example 1:**

Input: N = 5

Output: 120

**Example 2:**

Input: N = 4

Output: 24

**Sol:**

def factorial(N):

if N == 0:

return 1

fact = 1

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

fact \*= i

return fact

print(factorial(5)) # Output: 120

print(factorial(4)) # Output: 24

**Question 4**

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

**Sol:**

def calculateExponent(N, P):

return N \*\* P

print(calculateExponent(5, 2)) # Output: 25

print(calculateExponent(2, 5)) # Output: 32

**Question 5**

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

**Sol:**

def findMax(arr, n):

# Base case: if the array contains only one element

if n == 1:

return arr[0]

# Recursive case: divide the array into two halves

max1 = findMax(arr[:n//2], n//2)

max2 = findMax(arr[n//2:], n - n//2)

# Compare and return the maximum of the two halves

return max(max1, max2)

arr1 = [1, 4, 3, -5, -4, 8, 6]

print(findMax(arr1, len(arr1))) # Output: 8

arr2 = [1, 4, 45, 6, 10, -8]

print(findMax(arr2, len(arr2))) # Output: 45

**Question 6**

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

**Sol:**

def findNthTerm(a, d, n):

return a + (n - 1) \* d

print(findNthTerm(2, 1, 5)) # Output: 6

print(findNthTerm(5, 2, 10)) # Output: 23

**Question 7**

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”*

**Sol:**

def permuteString(string, left, right):

if left == right:

print(string)

else:

for i in range(left, right + 1):

string = swapCharacters(string, left, i)

permuteString(string, left + 1, right)

string = swapCharacters(string, left, i) # backtrack

def swapCharacters(string, i, j):

charList = list(string)

charList[i], charList[j] = charList[j], charList[i]

return ''.join(charList)

# Driver code

S = "ABC"

permuteString(S, 0, len(S) - 1)

ABC

ACB

BAC

BCA

CBA

CAB

S = "XY"

permuteString(S, 0, len(S) - 1)

XY

YX

**Question 8**

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

**Sol:**

def productOfArray(arr):

product = 1

for num in arr:

product \*= num

return product

arr1 = [1, 2, 3, 4, 5]

print(productOfArray(arr1)) # Output: 120

arr2 = [1, 6, 3]

print(productOfArray(arr2)) # Output: 18