ASSIGMNENT -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

SOLUTIONS:

TC:O(n), SC:O(1)

CODE:

```
class Solution:
    def isPowerOfTwo(self, n: int) -> bool:
        # If n <= 0 that means its a negative hence not a power of 2...
        if n <= 0:
            return False
        if n == 1:
            return True
        # Keep dividing the number by ‘2’ until it is not divisible by ‘2’
anymore.
        while (n % 2 == 0):
            n /= 2
        # If n is equal to 1, The integer is a power of two otherwise
false...
        return n == 1
```

OR

```
#or bitwise
      return (n != 0) and ((n & (n - 1)) == 0)
```

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

SOLUTIONS:

TC:O(n), SC:O(1)

CODE:

```
def findSum(n):
    sum = 0
    x = 1
    while x <= n:
        sum = sum + x
        x = x + 1
    return sum
```

OR

```
def findSum(n) :
    return n * (n + 1) / 2
n = 5
print(findSum(n))
```

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

SOLUTION: TC: O(n), SC: O(n) – as it's recursive approach

```
def factorial(n):
  if n==0 or n==1:
    return 1
  return n*factorial(n-1)

n = 5
print(factorial(n))
```

OR

```
def factorial(n): #TC:O(n), SC:O(1) -> iterative approach
    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))
```

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

SOLUTION: TC: O(n), SC(1)

```
def expo(N,P):
  return N**P

N, P = 5,2
```

```
print(expo(N,P))
```

OR

```
N = int(input("Enter the number:"))
P = int(input("Enter the power:"))
print(pow(N,P))
```

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

SOLUTION: TC:O(n), SC:O(1)

```
def findMaxRec(A, n):
    if (n == 1):
        return A[0]
    return max(A[n - 1], findMaxRec(A, n - 1))


A = [1, 4, 45, 6, -50, 10, 2]
n = len(A)
print(findMaxRec(A, n))
```

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

SOLUTION: TC: O(1), SC:O(1)

```
def printAP(a,d,n):

    # Printing AP by simply adding d
    # to previous term.
  return (a+(n-1)*d)


# Driver code
a = 5 # starting number
d = 2 # Common difference
n = 10 # N th term to be find

printAP(a, d, n)
```

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"

SOLUTION: TC:O(n), SC:O(n)

```
from itertools import permutations

words = [''.join(p) for p in permutations('pro')]

print(words)
```

OR

```
# Recursive function to generate all permutations of a string
def permutations(remaining, candidate=''):

    if len(remaining) == 0:
        print(candidate)

    for i in range(len(remaining)):

        newCandidate = candidate + remaining[i]
        newRemaining = remaining[0:i] + remaining[i+1:]

        permutations(newRemaining, newCandidate)


if __name__ == '__main__':

    s = 'ABC'
    permutations(s)
```

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

SOLUTION: TC:O(n), SC:O(1)

```
a = [1,2,3,4,5]
prod = 1
for i in a:

  prod = prod*i

print(prod)
```