

Assignment 09 Solutions

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 $n == 2^x$.

Example 1: Input: $n = 1$

Output: true

Example 2: Input: $n = 16$

Output: true

Example 3: Input: $n = 3$

Output: false

```
In [17]: def isPowerOfTwo(n):  
         if n <= 0:  
             return False  
         return (n & (n - 1)) == 0
```

```
In [18]: print(isPowerOfTwo(1))  
         print(isPowerOfTwo(16))  
         print(isPowerOfTwo(3))
```

True
True
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

```
In [19]: def sumOfNaturalNumbers(n):  
         return (n * (n + 1)) // 2
```

```
In [20]: print(sumOfNaturalNumbers(3))  
         print(sumOfNaturalNumbers(5))
```

6
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

```
In [21]: def factorial(N):  
         factorial = 1  
         for i in range(1, N + 1):  
             factorial *= i  
         return factorial
```

```
In [22]: print(factorial(5))
print(factorial(4))
```

```
120
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

```
In [23]: def calculateExponent(N, P):
        return N ** P
```

```
In [24]: print(calculateExponent(5, 2)) # Output: 25
print(calculateExponent(2, 5)) # Output: 32
```

```
25
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

```
In [25]: def findMax(arr):
        # Base case: If array contains only one element, return that element
        if len(arr) == 1:
            return arr[0]

        # Split the array into two halves
        mid = len(arr) // 2
        left = arr[:mid]
        right = arr[mid:]

        # Recursively find the maximum elements of the left and right halves
        max_left = findMax(left)
        max_right = findMax(right)

        # Compare the maximum elements of the left and right halves and return the larger one
        return max(max_left, max_right)
```

```
In [26]: arr1 = [1, 4, 3, -5, -4, 8, 6]
print(findMax(arr1)) # Output: 8

arr2 = [1, 4, 45, 6, 10, -8]
print(findMax(arr2)) # Output: 45
```

```
8
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

```
In [27]: def findNthTerm(a, d, N):  
         return a + (N - 1) * d
```

```
In [28]: print(findNthTerm(2, 1, 5)) # Output: 6  
         print(findNthTerm(5, 2, 10)) # Output: 23  
  
6  
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"

```
In [29]: def permute(S):  
         # Base case: If the length of S is 1, return the string itself as the only permutation  
         if len(S) == 1:  
             return [S]  
  
         # Set to store all permutations  
         permutations = []  
  
         # Generate permutations for each character in S  
         for i in range(len(S)):  
             # Remove character at index i from S  
             c = S[i]  
             S_without_c = S[:i] + S[i+1:]  
  
             # Recursively generate all permutations of S_without_c  
             sub_permutations = permute(S_without_c)  
  
             # Append c to each permutation  
             for sub_permutation in sub_permutations:  
                 permutations.append(c + sub_permutation)  
  
         return permutations
```

```
In [30]: S1 = "ABC"  
         print(permute(S1)) # Output: ['ABC', 'ACB', 'BAC', 'BCA', 'CAB', 'CBA']  
  
         S2 = "XY"  
         print(permute(S2)) # Output: ['XY', 'YX']  
  
['ABC', 'ACB', 'BAC', 'BCA', 'CAB', 'CBA']  
['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

```
In [31]: def productOfArray(arr):  
         product = 1  
         for num in arr:  
             product *= num  
         return product
```

```
In [33]: arr1 = [1, 2, 3, 4, 5]
```

```
print(productOfArray(arr1)) # Output: 120
```

```
arr2 = [1, 6, 3]
```

```
print(productOfArray(arr2)) # Output: 18
```

```
120
```

```
18
```

In []:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js