Day 1

1) Given a **sorted** array **arr[]** of distinct integers. Sort the array into a wave-like array(In Place). In other words, arrange the elements into a sequence such that arr[1] >= arr[2] <= arr[3] >= arr[4] <= arr[5]..... If there are multiple solutions, find the lexicographically smallest one.

Input:
n = 5
arr[] = {1,2,3,4,5}
Output: 2 1 4 3 5
Explanation: Array elements after sorting it in wave form are 2 1 4 3 5.

Input:
n = 6
arr[] = {2,4,7,8,9,10}
Output: 4 2 8 7 10 9
Explanation: Array elements after sorting it in wave form are 4 2 8 7 10 9.

2) You are given an array arr[] of size n. Find the total count of sub-arrays having their sum equal to 0.

Input:
n = 6
arr[] = {0,0,5,5,0,0}
Output: 6
Explanation: The 6 subarrays are [0], [0], [0], [0], [0,0], and [0,0].

Input:
n = 10
arr[] = {6,-1,-3,4,-2,2,4,6,-12,-7}
Output: 4
Explanation: The 4 subarrays are [-1 -3 4] [-2 2], [2 4 6 -12] & [-1 -3 4 -2 2]

3) Given an array arr[], its starting position l and its ending position r. Sort the array using merge sort algorithm.

Input:
N = 5
arr[] = {4 1 3 9 7}
Output:
1 3 4 7 9

Input:
N = 10
arr[] = {10 9 8 7 6 5 4 3 2 1}
Output:

1 2 3 4 5 6 7 8 9 10

4) Given a string S. The task is to print all unique permutations of the given string in lexicographically sorted order.

Input: ABC
Output:
ABC ACB BAC BCA CAB CBA
Explanation:
Given string ABC has permutations in 6 forms as ABC, ACB, BAC, BCA, CAB and CBA

Input: ABSG
Output:
ABGS ABSG AGBS AGSB ASBG ASGB BAGS BASG BGAS BGSA BSAG BSGA GABS GASB GBAS GBSA GSAB GSBA SABG SAGB SBAG SBGA SGAB SGBA
Explanation: Given string ABSG has 24 permutations.

5) Given an unsorted array, **Arr[]** of size **N** and that contains **even** number of occurrences for all numbers except two numbers. Find the two numbers in **decreasing** order which has **odd** occurrences.

Input:
N = 8
Arr = {4, 2, 4, 5, 2, 3, 3, 1}
Output: {5, 1}
Explanation: 5 and 1 have odd occurrences.

Input:
N = 8
Arr = {1 7 5 7 5 4 7 4}
Output: {7, 1}
Explanation: 7 and 1 have odd occurrences.

6) Given a string `s` that contains parentheses and letters, remove the minimum number of invalid parentheses to make the input string valid. Return *all the possible results*. You may return the answer in **any order**.

Input: s = "()())()"
Output: ["(())()","()()()"]

Input: s = "(a)())()"
Output: ["(a())()","(a)()()"]

Input: s = ")("

Output: [""]

7) You are given an unordered array consisting of consecutive integers [1, 2, 3, ..., n] without any duplicates. You are allowed to swap any two elements. Find the minimum number of swaps required to sort the array in ascending order.

Sample Input 0

```
4
4 3 1 2
```

Sample Output 0

```
3
```

Explanation 0 Given array 4 3 1 2

After swapping (0,2) we get (1, 3, 4, 2)
After swapping (1,2) we get (1, 4, 3, 2)
After swapping (1,3) we get (1, 2, 3, 4)
So, we need a minimum of 3 swaps to sort the array in ascending order.

8) Given a string `s`, reverse only all the vowels in the string and return it. The vowels are `'a'`, `'e'`, `'i'`, `'o'`, and `'u'`, and they can appear in both lower and upper cases, more than once.

Input: s = "hello"
Output: "holle"

Input: s = "uppercase"
Output: "epparcesu"

9) A **perfect number** is a **positive integer** that is equal to the sum of its **positive divisors**, excluding the number itself. A **divisor** of an integer `x` is an integer that can divide `x` evenly. Given an integer `n`, return `true` *if* `n` *is a perfect number, otherwise return* `false`.

Input: num = 28
Output: true
Explanation: 28 = 1 + 2 + 4 + 7 + 14
1, 2, 4, 7, and 14 are all divisors of 28.

Input: num = 7
Output: false

10) Given a non-negative integer `c`, decide whether there're two integers `a` and `b` such that $a^2 + b^2 = c$.

Input: c = 5
Output: true **Explanation:** 1 * 1 + 2 * 2 = 5

Input: c = 3
Output: false