Sr.	Problem Definition
1	WAP to find whether given number is **Prime or not**.
2	WAP to find whether given number is **Ugly or not**. An **ugly number** is a *positive* integer which does not have a prime factor other than 2, 3, and 5. **Input**: n = 6 **Output**: true **Explanation**: 6 = 2 × 3
3	WAP to find whether given number is **Kaprekar or not**. A **Kaprekar number** is a non-negative integer that, when squared, can be split into two parts whose sum equals the original number. For E.g. 45 is a Kaprekar number because 45 squared (2025) can be split into 20 and 25, and 20 + 25 = 45.
4	WAP to find whether given number is **Automorphic or not**. An **automorphic** number is a number whose square ends with the same digits as the number itself. For example, **5 is automorphic** because $5^2 = 25$, which ends in 5. Similarly, **76 is automorphic** because $76^2 = 5776$, which ends in 76.
5	WAP to find whether given number is **Pronic or not**. **A Pronic Number** is defined as a number that is the product of two consecutive non-negative integers. In other words, N is a Pronic Number if there exists a non-negative integer k such that N = k * (k + 1). E.g. **6 is a Pronic Number** because 6=2*3
6	Write a program to check whether number is **Happy number or not**. A **happy number** are those number whose digit's square summation eventually reaches to 1, if the sequence start repeating then it is not a happy number. E.g., **49 is a happy number** whose sequence is 49 97 130 10 1. E.g., **50 is not a happy number** whose sequence is 50 25 29 85 89 145 42 20 4 16 37 58 89 it should stop when 89 is detected 2nd time and print 50 is not a happy number.
7	WAP that **finds an angle** between clock hands for the provided value of **Minutes and Hours.**
8	WAP to Convert a **Decimal to Octal** and vice versa.
9	WAP to Convert a **Decimal to Hexa-decimal** and vice versa.
10	WAP to enter an element at specific position into **array**. **(Do not take a new array)**
11	Implement a following **pattern** For n=4 1 2 5 3 6 8 4 7 9 10 For n=5 1 2 6 3 7 10 4 8 11 13 5 9 12 14 15
12	WAP to implement a following **pattern** 1 1 * 1 1 * 3 * 1 1 * 3 * 5 * 3 * 1

Sr.	Problem Definition
13	Implement a following **pattern** *
14	Implement a following **pattern** 1 212 32123 4321234 32123 212 1
15	Implement a following **pattern** 1 1 2 1 1 * 3 * 1 1 * * 4 * * 1 1 * * * 5 * * * 1 1 * * * * 6 * * * * 1
16	Given the number of rows and columns, print the corresponding **swastika pattern using loops**. **Note:** The number of rows and columns should be the same and an odd number. This will generate a perfect swastika pattern. *
17	Write a program to print **Pascal triangle.**
18	Given two integer arrays **nums1 and nums2, return an array of their intersection**. Each element in the result must appear as many times as it shows in both arrays and you may return the result in sorted order.

<table>
<tr><th>Sr.</th><th>Problem Definition</th></tr>
<tr><td>19</td><td>Write a program to print following pattern
For n=5
1 2 3 4 5
10 9 8 7 6
11 12 13 14 15
20 19 18 17 16
21 22 23 24 25</td></tr>
<tr><td>20</td><td>Write a program to print following pattern
For n=4
<table>
<tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>4</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>3</td><td>2</td><td>2</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>3</td><td>2</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>3</td><td>2</td><td>2</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>3</td><td>3</td><td>3</td><td>3</td><td>3</td><td>4</td></tr>
<tr><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
</table></td></tr>
<tr><td>21</td><td>WAP to sort an Array using insertion sort.</td></tr>
<tr><td>22</td><td>WAP to sort an Array using selection sort.</td></tr>
<tr><td>23</td><td>Given an array, rotate the array to the right by k steps, where k is non-negative.
Input: nums = [1,2,3,4,5,6,7], k = 3
Output: [5,6,7,1,2,3,4]</td></tr>
<tr><td>24</td><td>Given an array arr[], the task is to reverse the array. Reversing an array means rearranging the elements such that the first element becomes the last, the second element becomes second last and so on. (Do not use any additional Array)
Input: arr[] = {1, 4, 3, 2, 6, 5}
Output: {5, 6, 2, 3, 4, 1}</td></tr>
<tr><td>25</td><td>WAP for following Scenario.
Given n rupees and a chocolate price of m for each chocolate, with a wrapper exchange offer of 1 chocolate per k wrappers, calculate the total number of chocolates you can eat with n rupees.</td></tr>
<tr><td>26</td><td>Find the difference between the second largest element and the second smallest element of an array.
Input : Enter the size of array: 7
Enter 7 elements: 5 1 9 7 1 5 3
Output: Difference: 4</td></tr>
<tr><td>27</td><td>WAP for given N digits, arrange them to form the largest number divisible by 3.</td></tr>
<tr><td>28</td><td>Write a program that should decode the given pattern and print the resulting expanded string.
Input: 2a3bc4dE5F2G7H
Output: aabbbcddddEFFFFFGGHHHHHHH</td></tr>
</table>

Sr.	Problem Definition
29	Given an **integer n**, return the least number of perfect square numbers that sum to n. **Input**: n = 12 **Output**: 3 **Explanation**: 12 = 4 + 4 + 4. **Input:** n = 13 **Output:** 2 **Explanation:** 13 = 4 + 9.
30	WAP to check whether number is present in array or not (using recursion only) and the function's syntax is given below **int isInArray(int a [], int m)**; Where int a [] is Array of integer and m is element to be searched.
31	Given a string **S** of length **n,** the task is to find the earliest repeated character in it. The earliest repeated character means, the character that occurs more than once and whose second occurrence has the smallest index. **Input**: s = "geeksforgeeks" **Output**: e **Explanation**: e is the first element that repeats
32	Given two strings s1 and s2 consisting of lowercase characters, the task is to check whether the two given strings are anagrams of each other or not. An anagram of a string is another string that contains the same characters, only the order of characters can be different. **Input**: s1 = "geeks" s2 = "kseeg" **Output**: true **Explanation**: Both the strings have the same characters with same frequency. So, they are anagrams.
33	Take an Input in the form of Binary String that contains only 0's and 1's and convert this number into integer. Input: 101.110 Output: 5.75
34	Given an array of positive integers arr[] of size n, the task is to find the second largest distinct element in the array. Note: If the second largest element does not exist, return -1. **Input**: arr[] = [12, 35, 1, 10, 34, 1] **Output**: 34 **Explanation**: The largest element of the array is 35 and the second largest element is 34. **Input**: arr[] = [10, 10, 10] **Output**: -1 **Explanation**: The largest element of the array is 10 there is no second largest element.
35	Given an array nums with n integers, your task is to check if it could become non-decreasing by modifying at most one element. **Input**: nums = [4,2,3] **Output**: true **Explanation**: You could modify the first 4 to 1 to get a non-decreasing array.
36	Given an array of N integers, and an integer K, find the number of pairs of elements in the array whose sum is equal to K. **Input**: N = 4, K = 6, arr[] = {1, 5, 7, 1} **Output**: 2

Sr.	Problem Definition
37	WAP to multiply two matrices with error checking for dimension compatibility.
38	Write a Function that returns either 1 or 0 based on following condition if the array is in ascending order and occurrence of that number at least 3 then it should return 1 otherwise it should return 0. e.g. **if A=[1,1,1,2,2] it should return 0, if A= [1,1,1,3,3,3,3] it should return 1.**
39	Implement the myAtoi(string s) function, which converts a string to a 32-bit signed integer. The algorithm for myAtoi(string s) is as follows: Whitespace: Ignore any leading whitespace (" "). Signedness: Determine the sign by checking if the next character is '-' or '+', assuming positivity is neither present. Conversion: Read the integer by skipping leading zeros until a non-digit character is encountered or the end of the string is reached. If no digits were read, then the result is 0. Rounding: If the integer is out of the 32-bit signed integer range [-231, 231 - 1], then round the integer to remain in the range. Specifically, integers less than -231 should be rounded to -231, and integers greater than 231 - 1 should be rounded to 231 - 1. Return the integer as the final result. Input: s = " -042" Output: -42 Explanation: Step 1: " -042" (leading whitespace is read and ignored) Step 2: " -042" ('-' is read, so the result should be negative) Step 3: " -042" ("042" is read in, leading zeros ignored in the result)
40	Given 2 sorted arrays **a[]** and **b[]**, each of size **n**, the task is to find the median of the array obtained after merging a[] and b[]. **Input**: a[] = [1, 12, 15, 26, 38], b[] = [2, 13, 17, 30, 45] **Output**: 16 **Explanation**: The middle two elements are 15 and 17, so median = (15 + 17)/2 = 16
41	You are given **n** disks placed on a starting rod (from), with the smallest disk on top and the largest at the bottom. There are three rods: the **starting** rod(from), the **target** rod (to), and an **auxiliary** rod (aux). You have to calculate the total number of **moves** required to transfer all **n** disks from the starting rod to the target rod, following these rules: **1.** Only one disk can be moved at a time. **2.** A disk can only be placed on top of a larger disk or on an empty rod. Return the number of moves needed to complete the task. **Input**: n = 3 **Output**: 7 **Explanation**: For N=3, steps will be as follows in the example and total 7 steps will be taken. move disk 1 from rod 1 to rod 3 move disk 2 from rod 1 to rod 2 move disk 1 from rod 3 to rod 2

Sr.	Problem Definition
	move disk 3 from rod 1 to rod 3 move disk 1 from rod 2 to rod 1 move disk 2 from rod 2 to rod 3 move disk 1 from rod 1 to rod 3
42	Given an integer n, return the n^{th} digit of the infinite integer sequence [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, …]. **Input**: n = 11 => **Output**: 0 **Explanation:** The 11^{th} digit of the sequence 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, ... is a 0, which is part of the number 10.
43	WAP for following Scenario. A child is running up a staircase with n steps and can hop either **1 step, 2 steps, or 3 steps** at a time. The task is to implement a method to count how many possible ways the child can run up the stairs.
44	Find a next lexicographical order string from a given string, if input is abc output is acb (2 conditions must be followed: the string must be greater than the inputted string and the outputted string must be the smallest string from all possible strings).
45	You are given a 0-indexed array nums consisting of positive integers. You can choose two indices i and j, such that i != j, and the sum of digits of the number nums[i] is equal to that of nums[j]. Return the maximum value of nums[i] + nums[j] that you can obtain over all possible indices i and j that satisfy the conditions. **Input**: nums = [18,43,36,13,7] **Output**: 54 **Explanation**: The pairs (i, j) that satisfy the conditions are: - (0, 2), both numbers have a sum of digits equal to 9, and their sum is 18 + 36 = 54. - (1, 4), both numbers have a sum of digits equal to 7, and their sum is 43 + 7 = 50. So the maximum sum that we can obtain is 54.
46	Given a non-negative integer x, compute and return the square root of x. Since the return type is an integer, the decimal digits are truncated, and only the integer part of the result is returned. **Note: You are not allowed to use any built-in exponent function or operator, such as pow(x, 0.5) or x ** 0.5.**
47	You are given several boxes with different colors represented by different positive numbers. You may experience several rounds to remove boxes until there is no box left. Each time you can choose some continuous boxes with the same color, remove them and get k * k points. Return the maximum points you can get. **Input**: boxes = [1,3,2,2,2,3,4,3,1] **Output**: 23 **Explanation:** [1, 3, 2, 2, 2, 3, 4, 3, 1] ----> [1, 3, 3, 4, 3, 1] (3*3=9 points) ----> [1, 3, 3, 3, 1] (1*1=1 points) ----> [1, 1] (3*3=9 points) ----> [] (2*2=4 points)
48	Given an unsorted integer array nums, return the smallest missing positive integer. You must implement an algorithm that runs in O(n) time and uses constant extra space. **Input**: nums = [1,2,0] **Output**: 3

Sr.	Problem Definition
49	Given an array nums of distinct integers, return all the possible permutations. You can return the answer in any order. **Input**: nums = [1,2,3] **Output**: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
50	Given a string that contains a special character together with alphabets (‘a’ to ‘z’ and ‘A’ to ‘Z’), reverse the string in a way that special characters are not affected. **Input**: a!!!b.c.d,e'f,ghi **Output**: i!!!h.g.f,e'd,cba **Input**: str = “Ab,c,de!$” **Output**: str = “ed,c,bA!$”
51	Given an array of **coins[]** of size **n** and a target value **sum**, where **coins[i]** represent the coins of different denominations. You have an **infinite supply** of each of the coins. The task is to find the **minimum** number of coins required to make the given value **sum**. If it is not possible to form the sum using the given coins, return **-1**. **Input**: coins[] = [25, 10, 5], sum = 30 **Output**: 2 **Explanation** : Minimum 2 coins needed, 25 and 5 **Input**: coins[] = [9, 6, 5, 1], sum = 19 **Output**: 3 **Explanation**: 19 = 9 + 9 + 1
52	Find if a given string can be represented from a substring by iterating the substring “n” times. **Input**: str = "abcabcabc" **Output:** true **Input:** str = "aabaabaabaab" **Output:** true **Input:** str = "abcdabc" **Output:** false