SAVEETHA SCHOOL OF ENGINEERING
SAVEETHA INSTITUTE OF MEDICAL AND SCIENCES
CHENNAI-602105
INSTITUTE OF ARTIFICIAL INTELLIGENCE AND DATA SCIENCE
DEPARTMENT OF DATA SCIENCE

Course code/Name: CSA08/ Python Programming

DAY 3 PROGRAMS

1. In daily share trading, a buyer buys shares in the morning and sells them on the same day. If the trader is allowed to make at most 2 transactions in a day, whereas the second transaction can only start after the first one is complete (Buy->sell->Buy->sell). Given stock prices throughout the day, find out the maximum profit that a share trader could have made.

Test Case:

1.Input: prices = [7,1,5,3,6,4]
 Output: 7
2.Input: prices = [7,6,4,3,1]
 Output: 0
3.Input: [10, 22, 5, 75, 65, 80]
 Output:87
4.Input: [2, 30, 15, 10, 8, 25, 80]
 Output:100
5.Input: [10, 22, 5, 75, 65, 80]
 Output:0

2. The Project manager has to submit the project within a week period. He didn't find the proper combinations of team members to work on the project, Help him in finding the possible combinations available.

Accept 3 digits and find all the combinations

Sample Input:

123

Sample Output:

123
132
213
231
312
321

Test Cases:

1. 789
2. 1456
3. -856
4. 1001
5. 555

3. Given an array of integers nums, return the number of good pairs.

A pair (i, j) is called good if nums[i] == nums[j] and i < j.

Test Cases:

1.Input: nums = [1,2,3,1,1,3]
Output: 4
Explanation: There are 4 good pairs (0,3), (0,4), (3,4), (2,5) 0-indexed.

2. Input: nums = [1,1,1,1]
Output: 6
Explanation: Each pair in the array are good.
3.Input: nums = [1,2,3]
Output: 0
4. nums=[4,5,1,3,7]
5.nums=[1,2,2,3]

4. Add Binary
Given two binary strings a and b, return their sum as a binary string.

- a and b consist only of '0' or '1' characters.
- Each string does not contain leading zeros except for the zero itself.

Test cases:
1.Input: a = "11", b = "1"
Output: "100"
2.Input: a = "1010", b = "1011"
Output: "10101"
3.a= “1111”, b= “1010”
4.a= “101101”, b= “1100”
5.a= “15” b= “45”

5. Given an array of integers where each element represents the max number of steps that can be made forward from that element. Write a function to return the minimum number of jumps to reach the end of the array (starting from the first element). If an element is 0, they cannot move through that element. If the end isn’t reachable, return -1.

Test Case:

1. Input: arr[] = [1, 3, 5, 8, 9, 2, 6, 7, 6, 8, 9]
   Output: 3 (1-> 3 -> 9 -> 9)
2. Input: arr[] = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
   Output: 10
3. Input: arr[] = [2,3,1,1,4]
   Output: 2
4. Input: arr[] = [1, 3, 6, 1, 0, 9]
   Output: 3
5. Input: arr[] = [2,3,0,1,4]

Output: 2

6. Raju, has again started troubling people in your city. The people have turned on to you for getting rid of Raju. Raju presents to you a number consisting of numbers from 0 to 9 characters. He wants you to reverse it from the final answer such that the number becomes Mirror number. A Mirror is a number which equals its reverse. The hope of people are on you so you have to solve the riddle. You have to tell if some number exists which you would reverse to convert the number into Mirror
Sample input:
Enter the number: 123456
Sample output:
Mirror image: 654321
Test cases:

1. Sell123

2. 5489236
3. Abc-abc
4. %$$$$^&
5. -123456

7. Permutations

Given a collection of numbers, nums, that might contain duplicates, return all possible unique permutations in any order.

Test cases:

1.Input: nums = [1,1,2]
Output:
[[1,1,2],
 [1,2,1],
 [2,1,1]]
2.Input: nums = [1,2,3]
Output: [[1,2,3],[1,3,2],[2,1,3],[2,3,1],[3,1,2],[3,2,1]]
3.nums=[2,8,9,10]
4.nums=[-1,0,1]
5.nums=[1,1,1]

8. Given an array of strings strs, group **the anagrams** together. You can return the answer in **any order**.
An **Anagram** is a word or phrase formed by rearranging the letters of a different word or phrase, typically using all the original letters exactly once.
strs[i] consists of lowercase English letters.

Test Cases:

1.Input: strs = ["eat","tea","tan","ate","nat","bat"]
Output: [["bat"],["nat","tan"],["ate","eat","tea"]]
2.Input: strs = [""]
Output: [[""]]
3.Input: strs = ["a"]
Output: [["a"]]
4.strs= “banana”
5.strs= 12345

9. Given an input string s and a pattern p, implement regular expression matching with support for '.' and '*' where:

- '.' Matches any single character.
- '*' Matches zero or more of the preceding element.

The matching should cover the **entire** input string (not partial).

Test case:

1.Input: s = "aa", p = "a"
Output: false
2.Input: s = "aa", p = "a*"
Output: true
3.Input: s = "ab", p = ".*"
Output: true
4.Input: s = " aaa", p = "aa"
Output: true

5.Input: s = "aab", p = "c*a*b"
Output: true

10. Given two strings word1 and word2, return the minimum number of operations required to convert word1 to word2.

You have the following three operations permitted on a word:

- Insert a character
- Delete a character
- Replace a character

Test case:

1.Input: word1 = "horse", word2 = "ros"
Output: 3
2.Input: word1 = "intention", word2 = "execution"
Output: 5
3.Input: str1 = “sunday”, str2 = “saturday”
Output: 3
4.Input: str1 = “cat”, str2 = “cut”
Output: 1
5.Input: str1 = “girl”, str2 = “grill”
Output: 2