ANNAMACHARYA INSTITUTE OF TECHNOLOGY & SCIENCES, RAJAMPET
ML MANIA CODING COMPETITION 2K25 (LEVEL – 2)
Languages Supported: C, Java, Python

Date: 10 -04 -2025   **Duration: 60 min**   **Max. Marks: 100**

Instructions:
- Total number of questions are 5 and each question carries 20 marks.
- Marks will be awarded based on structure, implementation and effectiveness of your code.
- Each question contains 3 test cases.
- Participant who submits more questions with proper execution will be considered as priority.
- Browser usage is strictly prohibited.
- No external help or collaboration is allowed during the competition.
- Use the corresponding IDE for respective languages.
    1. C – Dev C++
    2. Python – PyCharm or Python IDLE
    3. Java – Notepad++

PROBLEM STATEMENTS

1. Happy Number

Write an algorithm to determine if a number n is happy.

A **happy number** is a number defined by the following process:
- Starting with any positive integer, replace the number by the sum of the squares of its digits.
- Repeat the process until the number equals 1 (where it will stay), or it **loops endlessly in a cycle** which does not include 1.
- Those numbers for which this process **ends in 1** are happy.

Return true *if* n *is a happy number, and* false *if not*.

Test Case 1:
Input: n = 19
Output: true
Explanation:
$1^2 + 9^2 = 82$
$8^2 + 2^2 = 68$
$6^2 + 8^2 = 100$
$1^2 + 0^2 + 0^2 = 1$

Test Case 2:
Input: n = 2
Output: false
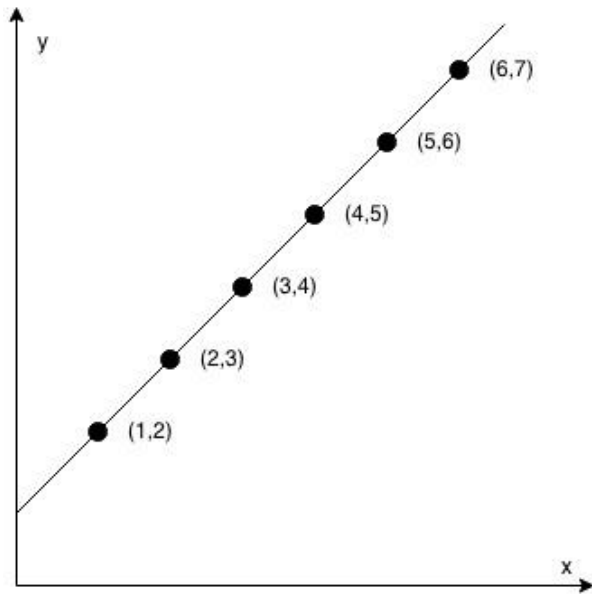
Test Case 3:
Input: n = 7
Output: true
Constraints:
- $1 <= n <= 2^{31} - 1$

2. Check if it is a Straight Line

You are given an array coordinates, coordinates[i] = [x, y], where [x, y] represents the coordinate of a point. Check if these points make a straight line in the XY plane.
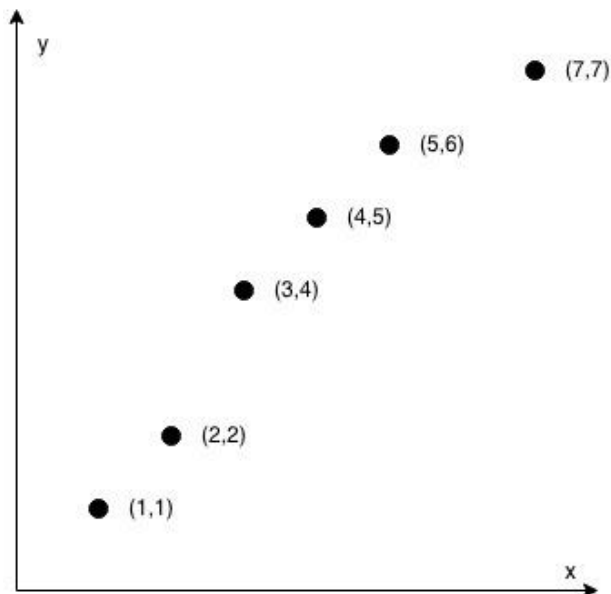
Test Case 1:

Input: coordinates = [[1,2],[2,3],[3,4],[4,5],[5,6],[6,7]]
Output: true

Test Case 2:

Input: coordinates = [[1,1],[2,2],[3,4],[4,5],[5,6],[7,7]]
Output: false

Test Case 3:
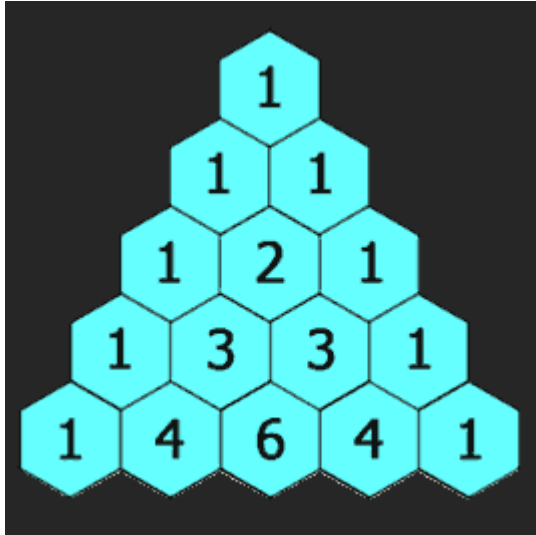**Input:** coordinates = [[2,2],[3,4],[4,5]]
Output: false

Constraints:
- 2 <= coordinates.length <= 1000
- coordinates[i].length == 2
- -10^4 <= coordinates[i][0], coordinates[i][1] <= 10^4
- coordinates contain no duplicate point.

3.Pascal's Triangle

Given an integer numRows, return the first numRows of **Pascal's triangle**.
In **Pascal's triangle**, each number is the sum of the two numbers directly above it as shown:

Test Case 1:
Input: numRows = 5
Output: [[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]
Test Case 2:
Input: numRows = 1
Output: [[1]]
Test Case 3:
Input: numRows = 0
Output: []
Constraints:
- 1 <= numRows <= 30

4.Two Sum

Given an array of integers 'nums' and an integer 'target', return *indices of the two numbers such that they add up to target*.
You may assume that each input would have **exactly** **one solution**, and you may not use the *same* element twice.
You can return the answer in any order.

Test Case 1:
Input: nums = [2,7,11,15], target = 9
Output: [0,1]
Explanation: Because nums[0] + nums[1] == 9, we return [0, 1].
Test Case 2:
Input: nums = [3,2,4], target = 6
Output: [1,2]
Test Case 3:
Input: nums = [3,3], target = 6
Output: [0,1]

Constraints:
- 2 <= nums.length <= 10^4
- -10^9 <= nums[i] <= 10^9
- -10^9 <= target <= 10^9

5.Median of Two Sorted Arrays

Given two sorted arrays nums1 and nums2 of size m and n respectively, return **the median** of the two sorted arrays.

The overall run time complexity should be O(log (m+n)).

Test Case 1:
Input: nums1 = [1,3], nums2 = [2]
Output: 2.00000
Explanation: merged array = [1,2,3] and median is 2.

Test Case 2:
Input: nums1 = [1,2], nums2 = [3,4]
Output: 2.50000
Explanation: merged array = [1,2,3,4] and median is (2 + 3) / 2 = 2.5.

Test Case 3:
Input: nums1 = [0, 0], nums2 = [0, 0]
Output: 0.00000

Constraints:
- nums1.length == m
- nums2.length == n
- 0 <= m <= 1000
- 0 <= n <= 1000
- 1 <= m + n <= 2000
- -10^6 <= nums1[i], nums2[i] <= 10^6