- munna's academy programming contest is a timed event that brings together budding programmers and problem solvers from various educational institutions to compete, learn, and grow. This contest is designed for students who are passionate about coding and eager to test their skills in a challenging environment.

MAPC | munna's academy programming contest

A. Candy Sharing Puzzle

Problem:
Three friends, Alice, Bob, and Charlie, are sharing a bag of candies. Write a program to calculate if the candies can be evenly distributed among them. If there are leftover candies, find how many are left.
Input:
The total number of candies as an integer.
Output:
Print "Evenly distributed" if the candies can be evenly divided among three friends. Otherwise, print the leftover candies.
Test Cases:

- Input: `15`
  Output: `Evenly distributed`
- Input: `17`
  Output: `Leftover candies: 2`

B. Distance Between Two Points

Read the four values corresponding to the x and y axes of two points in the plane, p1 (x1, y1) and p2 (x2, y2), and calculate the distance between them, showing four decimal places, according to the formula:

Distance = $\sqrt{(x2 - x1)^2 + (y2 - y1)^2}$

Input

The input file contains two lines of data. The first one includes two double values: ***x1 y1*** and the second one also contains two double values with one digit after the decimal point: ***x2 y2***.

Output

Calculate and print the distance value using the provided formula, with 4 decimal places.

Input Sample	Output Sample
1.0 7.0 5.0 9.0	4.4721
-2.5 0.4 12.1 7.3	16.1484

C. Prime Check

Problem:
Write a program to check if a number N is prime.

Input:
A single integer N.

Output:
Print "Prime" or "Not Prime."

Test Cases:
Input: 7
Output: Prime
Input: 15
Output: Not Prime

D. Find the Missing Number

Problem:
A list of numbers from 1 to N is missing one number. Write a program to find the missing number.

Input:
An integer N and a list of N−1 integers.

Output:
Print the missing number.

Test Cases:
Input: 5 [1, 2, 3, 5]
Output: 4
Input: 7 [2, 3, 4, 5, 6, 7]
Output: 1

E. Robot Movement on a Grid

Problem:

A robot starts at position (0,0) on a grid. It receives a string of instructions:
U moves up by 1 unit,
D moves down by 1 unit,
L moves left by 1 unit,
R moves right by 1 unit.

Calculate the robot’s final position after executing all instructions.

Input:

A string representing the robot's movement instructions.

Output:

The final coordinates (x,y).

Test Cases:

Input: UDLR
Output: (0, 0)
Input: UUUDDDDRRR
Output: (3, -1)