Task 1: Map Coloring (3-region map)

Problem:

You have 3 countries: A, B, and C.

A borders B and C

B and C do not border each other

You must color the map using 3 colors (Red, Green, Blue), with no two adjacent regions having the same color.

What to do:

List variables

Assign domain

Define constraints (binary)

**Solution:-**

Variables

Each country is a variable representing a region on the map:

* **A**
* **B**
* **C**

Domain

Each variable can take one of the three colors:

* **Domain(A) = {Red, Green, Blue}**
* **Domain(B) = {Red, Green, Blue}**
* **Domain(C) = {Red, Green, Blue}**

Binary Constraints

These specify which variables cannot have the same value (i.e., no two adjacent countries should share the same color).

* **A ≠ B** → A and B are adjacent; they must be different colors.
* **A ≠ C** → A and C are adjacent; they must be different colors.
* **B and C** → No constraint here, since they don’t share a border.

You’ve basically set up a simple CSP with 3 variables, equal domains, and two binary inequality constraints. It’s totally solvable with even just two colors—so having three colors makes it extra flexible

Task 2: Simple Sudoku (2x2 mini-grid)

Problem:

You are given a 2x2 Sudoku grid with 4 cells: A1, A2, B1, B2.

Each must contain a digit from 1 to 2

No digit repeats in any row or column

What to do:

Define the variables

What is the domain of each cell?

Write all constraints (all-diff style)

**Solution:-**

Variables

The grid consists of four cells:

* **A1**, **A2** (Top row)
* **B1**, **B2** (Bottom row)

🔢 Domains

Each cell can contain a digit from **1 to 2**, so:

* **Domain(A1) = {1, 2}**
* **Domain(A2) = {1, 2}**
* **Domain(B1) = {1, 2}**
* **Domain(B2) = {1, 2}**

🧠 Constraints (All-Diff Style)

We need to ensure no repeated digit in any **row** or **column**.

**Row Constraints:**

* **AllDiff(A1, A2)** → Top row digits must differ
* **AllDiff(B1, B2)** → Bottom row digits must differ

**Column Constraints:**

* **AllDiff(A1, B1)** → First column digits must differ
* **AllDiff(A2, B2)** → Second column digits must differ

Task 3: Exam Scheduling

Problem:

You must schedule exams for 3 subjects: Math, English, and Science.

Each exam must be scheduled in one of 2 time slots: Morning, Afternoon

The same teacher teaches Math and Science, so those exams cannot be at the same time

What to do:

Variables = subjects

Domains = time slots

Constraint = binary (Math ≠ Science)

**Solution:-**

Variables

Each subject represents a variable:

* **Math**
* **English**
* **Science**

Domains

Each variable can be assigned one of two time slots:

* **Domain(Math) = {Morning, Afternoon}**
* **Domain(English) = {Morning, Afternoon}**
* **Domain(Science) = {Morning, Afternoon}**

Binary Constraint

We need to prevent scheduling Math and Science at the same time:

* **Math ≠ Science**

This binary inequality constraint ensures that the teacher who handles both subjects won't have to be in two places at once.

Task 4: Cryptarithmetic Puzzle (SEND + MORE = MONEY)

Problem:

In the puzzle SEND + MORE = MONEY, each letter stands for a unique digit from 0–9.

No leading digit (S or M) can be 0.

What to do:

Identify all letter variables

State domain for each

Define constraints:

AllDiff

Arithmetic equation holds

S ≠ 0, M ≠ 0

**Solution:-**

Variables

Each letter stands for a distinct digit:

* **S, E, N, D, M, O, R, Y**

That’s **8 unique variables**.

🔢 Domains

Each variable can take on values from 0 to 9 **except for S and M**, which cannot be 0 (no leading zeros allowed).

* **Domain(S) = {1–9}**
* **Domain(M) = {1–9}**
* **Domain(E, N, D, O, R, Y) = {0–9}**

🧠 Constraints

✅ All-Diff Constraint

All letters must represent **different** digits:

* **AllDiff(S, E, N, D, M, O, R, Y)**

🚫 No Leading Zeros

* **S ≠ 0**

 **M ≠ 0**

➕ Arithmetic Equation Constraint

The heart of the puzzle:

SEND + MORE = MONEY

We interpret each word as a numerical expression:

* **SEND = 1000×S + 100×E + 10×N + D**
* **MORE = 1000×M + 100×O + 10×R + E**
* **MONEY = 10000×M + 1000×O + 100×N + 10×E + Y**

So the final equation becomes:

(1000×S + 100×E + 10×N + D) + (1000×M + 100×O + 10×R + E) = (10000×M + 1000×O + 100×N + 10×E + Y)