**Background and Questions**

Background of the Problem

SET is a card game based on pattern recognition. The goal is to find a "set" of three cards that follow specific rules. Each card has four attributes: number, shape, color and shading, with three possible values for each.

A valid set consists of three cards where, for each attribute, the values are either all the same or all different. Mathematically, a SET card can be represented as a list with four values in {0,1,2}. A set is a group of three such vectors where each attribute follows the set rules.

In this project, we developed a Python program where a player competes against a computer in SET. There are always 12 cards on the table. The first to find a set scores a point. If the player does not find a set within a time limit, the computer gets a point and immediately reveals the set.

## **Challenges in Building the Game**

### Structuring the code and learning pygame

One of our first challenges is figuring out how to structure the game using classes and functions. We also need to decide how to create classes for the cards, manage the deck, and handle user input.   
Eventually, we need the visualize the game. Therefore, we need pygame and since neither of us used pygames before, this is going to be a challenge.

### **Creating the computer opponent**

A challenge is designing a code in python that finds sets instantly. The computer must be able to quickly scan the 12 cards and detect a valid set. However, it should not reveal it right away. Instead, if the player doesn’t find a set in time, the computer should display the set and score a point. Keeping track of the score and showing it clearly on the screen is another challenge we must solve.

### **Representing cards and checking for sets**

We need to create a class for the cards that stores their four properties. The program also needs a function that checks if three chosen cards form a valid set. To keep the game fair, we must shuffle and draw cards from the deck in a way that prevents predictable patterns.

### **Designing the game interface**

The game needs to be easy to play and understand. We must decide how to show the 12 cards clearly, how the player will select cards, and how to give feedback when a set is correct or incorrect. With Pygame, we need to figure out how to highlight selected cards and update the game screen smoothly.

### **Implementing buttons and player interaction**

To make the game easy to play, we need buttons for selecting cards and submitting a set. We must also add visual effects to show when a button is selected or disabled.

### **Optimizing the game for speed**

Since the game has to check many card combinations, we need to make sure our code runs efficiently. The goal is to make the game smooth and quick. Understanding how much time and memory our set-checking algorithm uses will help us improve performance.

By solving these challenges step by step, we aim to create a well-structured and fun SET game in Python that gives players a exciting challenge against the computer.