1. Possible combinations:

<u>* Case 1: (Lowercase only)</u>

Each segment = ΔΦΔ

= 21 x 5 x 21

= 2205

Two segments = ΔΦΔ ΔΦΔ

= 2205 x 2205

= **4.862.025** possible passwords

Hence, there are **4.862.025** possible passwords the generator can generate.

<u>*Case 2: (with Case Sensitivity – Lowercase and Uppercase)</u>

Each segment = ΔΦΔ

= 42 x 10 x 42

= 17640

Two segments = ΔΦΔ ΔΦΔ

= 17640 x 17640

= **311.169.600** possible passwords

Hence, there are **311.169.600** possible passwords the generator can generate.

2. Password: Alice7894508 n = 5 Challenge(n-c)

H^1(Alice7894508) = 34708019735e851c631eb33a6a3c685e

H^2(H^1(Alice7894508)) = **ef744d8bb177c08c25bb0933ee197e78**

H^3(H^2(H^1(Alice7894508))) = **ac25c178eb72e8ae623710f5114126f5**

H^4(H^3 H^2(H^1(Alice7894508)))) = **e9d1b90f4c1ba1f79d45043acdcba17e**

Hence, first 3 OTP transmitted by Alice:

		One-Time Password
1.	H^4(H^3 H^2(H^1(Alice7894508))))	e9d1b90f4c1ba1f79d45043acdcba17e
2.	H^3(H^2(H^1(Alice7894508)))	ac25c178eb72e8ae623710f5114126f5
3.	H^2(H^1(Alice7894508))	ef744d8bb177c08c25bb0933ee197e78

3. a) - No, the diagram does not define a lattice because it has no greatest lower bound that is being dominated by all elements which violates the definition of Lattice.

- Moreover, in a lattice, each element must be unique within the structure. However, the diagram shows two identical elements labeled with ‘ T ‘.

Hence, the diagram does not define a lattice as it violates the definition of a Lattice.

b) Assuming the diagram has been corrected, there still exists redundancy or we called it Transitive relationships which are:

1. **Z→R**:

   There’s a path from Z to R through Z→S→R.
   Hence, Z→R is redundant.

2. **S→P**:

   There’s a path from S to P through S→Q→P.
   Hence, S→P is redundant.

4. a) **Subjects** = Alexis, Boris, Catherine, Duggy the Dog
   Objects = Balls, Sticks, Boris
   Actions = Kick, Throw, Catch, Snap, Roll, Chew, Fetch

   b) **Access Control Matrix**

Subjects\Objects	**Balls**	**Sticks**	**Boris**
Alexis	Kick	Throw	
Boris	Catch, Kick	Throw	
Catherine	Roll	Snap	
Duggy the Dog	Chew	Fetch	Chew

c) **Access Control List**

Balls = (Alexis, Kick), (Boris, Catch/Kick), (Catherine, Roll), (Duggy the Dog, Chew)

Sticks = (Alexis, Throw), (Boris, Throw), (Catherine, Snap), (Duggy the Dog, Fetch)

Boris = (Duggy the Dog, Chew)

d) **Capabilities**

Alexis = (Balls, Kick), (Sticks, Throw)

Boris = (Balls, Catch/Kick), (Sticks, Throw)

Catherine = (Balls, Roll), (Sticks, Snap)

Duggy the Dog = (Balls, Chew), (Sticks, Fetch), (Boris, Chew)

Part Two: Two factor Authentication

Explanation of the Relationship:

Credential Storage (Passwords.txt):

- This file is used to store user credentials in a simple format where each line contains a username and its corresponding password. The structure allows for easy retrieval and verification during the authentication process.

User Registration and Validation (connect.py):

- Manages user registration and validation. Reads username and password and checks these against the stored credentials in Passwords.txt. If the user wants to register, it writes (stores) the new username and password into Passwords.txt.
- If user attempts to log in, connect.py verifies the credentials with those stored in the Passwords.txt file. This establishes user authentication by ensuring only registered users with correct credentials can proceed.
- Utilization of the OTP generated by device.py as a PIN for user login.

OTP Generator (device.py):

- It can generate a one-time password (OTP) using the device.py file. The OTP is calculated based on the username, password, and the current time.

Why the PIN Values Don't Leak Password Information:

One-Way Hashing: The OTP (One-Time Password) is created using a hashing method that transforms the username, password, and current time into a unique code. This process is one-way, which means it's nearly impossible to figure out the original username or password just by looking at the OTP.

Time-Sensitive: The OTP changes every 15 seconds based on the current time. So, even if someone manages to get an OTP, it only works for a short time, making it less useful for attackers.

Extra Security with PIN: The PIN is separate from the password, adding another layer of security. Even if someone knows your password, they still need the correct PIN to access your account. This makes it harder for anyone to get in without full credentials.

Hard to Guess: Because the OTP is generated based on the current time and requires both the username and password, it's really difficult for someone to guess it. An attacker would need to know your username, password, and the exact time to generate a valid OTP.