**Implementation of Random number generation using a subset of digits and alphabets**

**AP25122050012**

**Ramanadham Shiny**

**Random number generator (RNG):**

Random number generator is an algorithm or principle that generates random value with combination of alphabet, numbers which is unpredictable.

There are three types of Random number generators are there.

1. True Random Number Generator(TRNG)
2. Pseudo-Random Number Generator(PRNG)
3. Cryptographically Pseudo-Random Number Generator(CPRNG)
4. Quasi-Random Generator

As per problem statement we are implementing Pseudo-Random Number Generator. It mainly focused on general random generation and not mentioned security, unpredictable and cryptography. This is typically solved using random module in python, which is deterministic and repeatable if seeded.

**Limitations:**

* This is not secure.
* The output can be predicted if pattern is known
* Not suitable for cryptographic tasks (key generation or secure tokens)

**Code :**

import random

def generate\_random\_string(length, digits\_subset='0123456789', alphabets\_subset='abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ'):

"""

Generates a pseudo-random string using a subset of digits and alphabets.

Args:

length (int): Length of the string to generate.

digits\_subset (str): Allowed digits (e.g., '13579').

alphabets\_subset (str): Allowed alphabets (e.g., 'abcXYZ').

Returns:

str: Randomly generated string.

"""

pool = digits\_subset + alphabets\_subset

if not pool:

raise ValueError("Character pool cannot be empty.")

return ''.join(random.choice(pool) for \_ in range(length))

# Example usage

if \_\_name\_\_ == "\_\_main\_\_":

digits = '13579' # odd digits

alphabets = 'abcXYZ' # custom letters

length = 10

result = generate\_random\_string(length, digits, alphabets)

print("Pseudo-random string:", result)

**Output:**

