CRYPTACOUNT

DISCRETE PASSWORD ENTROPY ANALYZER

BY: KHARL DENZELL D. BUGARIN

INTRODUCTION & PROBLEM SCENARIO

How do we measure the strength of a password?

A password's security is measured by its Entropy (strength in bits). This is determined by the total number of unique password possibilities (the Search Space, N).

Topics Implemented

1. Permutations & Combinations

2. Rules of Sum & Product

Project Goal

To use discrete mathematics principles to precisely calculate the search space N under specific, complex user constraints.

The Program's Role

Functions as both a Calculator (to find N) and a Generator (to provide a valid sample Permutation).

MATHEMATICAL CONCEPTS: THE FOUNDATION

The Rule of Product

Purpose: Calculates total possibilities when choices are sequential or independent. This is the source of exponential growth in passwords.

Formula Used: Total N is the product of three independent counting factors.

Topic 2: Permutations & Combinations

Permutation: The final generated password (e.g., aB1c2D!E) is a unique ordered arrangement—a Permutation—from the total set.

Combination (The Core Logic): We use a generalized combination formula (the Multinomial Coefficient) to determine the number of ways to arrange the types of characters within the password slots.

This formula calculates L!/k1! * k2! * * * * r! to solve the complex placement problem.

TECHNICAL IMPLEMENTATION

Language

Python (using math for factorials and secrets for secure generation).

Calculator (calculate_search_space)

Input: Length (L), Required Counts (k).

Process: Calculates (1) the Multinomial Coefficient, (2) the Required Fill Choices, and (3) the Remaining Fill Choices.

Generator (generate_password)

Process: Randomly selects characters, ensures the k required characters are included, and shuffles the list (creating a valid Permutation).

Entropy Measure

The program converts N into a real-world metric: Entropy (H) in bits, using log2(N).

Scenario

Password Length: 8

Included: Lower, Upper, Digits

Constraint: Must contain EXACTLY 2 Digits (k=2).

LIVE DEMO

Mathematical Proof

Full Formula: $N = (L! / k! * r!) * (C\ req)^k * (C\ other)^r$

Factor 1: 8! / 2! * 6! = 28 (Multinomial Coefficient): The number of ways to choose 2 positions for digits out of 8 total slots (Combinations).

Factor 2: $10^2 * 52^6$ (Rule of Product): Multiplies the 28 position choices by the actual number of ways to fill those slots with characters (10 choices for the 2 digit spots, 52 choices for the 6 remaining spots).

By multiplying these independent factors, we correctly count every single possible unique password (N).

CONCLUSION & REFLECTION

Conclusion

CryptaCount successfully proved that discrete mathematics, particularly the Rule of Product, is the fundamental driver of security, demonstrating the exponential nature of cryptographic keys.

Learning

The project reinforced the skill of translating a real-world problem into a precise, multi-step mathematical model using Permutations and Combinations.

Conclusion

The main challenge was implementing the Multinomial Coefficient and ensuring the logic for calculating the available pool of 'other characters' (C_{other}) was perfectly mutually exclusive with the required set.

THANK YOU!