Computational Project – Classical Counting

Permutations, Combinations, r-Permutations, r-Combinations, and Bars and Stars

(Write the group names here)

(delivery date)

Document structure. This file contains two parts:

1. The problems (what the student must submit).
2. The solution (answers and justifications).

PART I — The Problems (slightly simplified version)

The goal is to implement and test the classic counting formulas in small cases:

Permutations n!, P(n, r) = (n ÿ r)!, $\frac{n!}{r}$, n^r, $\binom{n+r-1}{r}$, $\#\{x \in \mathbb{Z}^k_{\geq 0} : x_i = m\} = \binom{m+k-1}{k-1}$.

Delivery (easier)

- A single Count.py file with functions:

  factorial(n), nPr(n, r), nCr(n, r), nPr_rep(n, r), nCr_rep(n, r), star_bars(m, k).

- A block of if __name__=="__main__" that prints the results of the exercises in item (c) and, if desired, runs minimal tests.
- Basic error handling: If parameters are invalid (e.g., r > n in nCr or nP r), return 0 or throw ValueError with a short message (choose one convention and be consistent).
- Boundary exploration (required, brief). Implement a simple experiment that shows how far your implementation goes before enumeration becomes slow:

  i) Define a time threshold T (suggested: T = 2 s). ii) For each operation with possible enumeration (P(n, r) without repetition, ny stars), increment r, $\binom{n}{r}$, $\binom{n+r-1}{r}$, bars parameters from small values (e.g. n = 2, 3, ...) and stop when the enumeration time exceeds T.

  iii) Print one line per operation with the largest tested size that was below T (e.g.: P(n,r) enum OK until n=8,r=4; 1.7s).

  iv) Compare with the time of the closed formula (which should be practically instantaneous) and comment on it in one line.

- Optional but recommended: a test per operation that checks by enumeration a small case (e.g. n ÿ 6, r ÿ 4, m ÿ 8, k ÿ 4) and prints OK if it matches.

Problems to solve

(a) Implementation. Implement the six functions using closed formulas.

(b) Brief justification. In 5–8 lines (or comments) explain a combinatorial idea by formula (no lengthy demonstration required).

(c) Numerical exercises. Print the values:

$$6! = ?$$

$$P(7, 3) = ? 10$$

$$= ?$$
$$4$$

$$^{4\,5} = ?$$

$$3 + 5 \ddot{y} 1 \quad = ?$$
$$5$$

$$\#\{(x1, x2, x3) \ddot{y} Z\ddot{y}0 : x1+x2+x3 = 8\} = ?$$

(d) Minimal verification (optional). For one case per operation, compares formula vs. enumeration and displays OK.

Evaluation criteria (20 pts)

1. Correctness of formulas and impressions (8 pts).

2. Code quality and basic error handling (5 pts).

3. Exploration of limits: simple design, clear output and brief commentary (5 pts).

4. Brief justification and clarity (2 pts).

5. Optional minimum verification (+2 extra points).
