Name Numerology Analyzer

Language: C (ISO C11)

Objective: Read one or more names, compute numerology numbers using letter→number mapping (A=1 … Z=26), optionally reduce to a core number (single digit, preserving master numbers 11 and 22), store results in memory and optionally save to CSV. Program must be robust: handle invalid inputs, buffer overflows, file errors, and all edge cases.

Features implemented

- Interactive menu (add name, view all results, save to CSV, exit).
- Input validation and cleaning (removes non-alphabet chars; preserves only A–Z).
- Case-insensitive calculation (A == a).
- Long-line handling with trimming and warning.
- Dynamic storage of results (array of `struct` entries grown with `realloc`).
- Safe file operations with error checks.
- Memory-safety practices: bounds checking, null-termination, freeing memory.
- Test cases and checks for all common edge cases.

Assumptions / Configurable rules

- Letter mapping: A=1, B=2, ..., Z=26.
- Reduction: sums can be reduced to a single digit (1–9) unless they are master numbers 11 or 22, which are preserved. (If your teacher expects a different reduction rule, this is trivial to switch.)

Textual flowchart

Start → Display menu
├─ 1) Add name → Read line → Trim → If overflow warn; Clean (keep letters) → If no letters reject → Compute numerology → Reduce → Append to array → return to menu
├─ 2) View results → Print a formatted table of stored `Person` entries → return to menu
├─ 3) Save to CSV → Attempt fopen → If error report → Else write CSV rows

→ close → return to menu
└── 4) Exit → Free memory → End