Congratulations! **1-Hour Software** has put you in charge of the Plan Rank module (PRANK) of their Planning Assistant system. PRANK has to say whether one plan is better or worse than another plan, for a given set of prioritized goals and knowledge of how the plans affect those goals

Specifically, PRANK will get

- Goals (names only) organized into one or more groups of equal importance, e.g., [A B]; [C]; [D E F]; [G H] if A and B are equally important and more important than C, which in turn is more important than D, E and F, which are equivalent and more important than G and H.
- Plans (names only) and the goals, if any, they positively and negatively affect, e.g., [P1 [+ B E F H] [- C D G]]; [P2 [+ C D F] [- E H]] if P1 is positive for goals B, E, F and H and negative for C, D and G, and P2 is positive for C, D, and F, and negative for E and H.

A plan is better than another plan for a specific goal if it's

- positive for that goal and the other plan is neutral or negative, or
- neutral for that goal and the other plan is negative.

A plan is better than another plan if, over all goals,

- it is better for at least one goal than the other plan, and
- there is no more important goal that the other plan is better for.

Define a function COMPARE-PLANS(*plan1*, *plan2*, *goal-data, plan-data*) to take the names of two plans, and data about goals and plans, and return:

- BETTER if *plan1* is better than *plan2*,
- WORSE if it's worse,
- MIXED if it's better for some goals, but the other plan is better for other equally important goals
- SAME if the plans don't differ

Use any programming language and data structures you like to implement PRANK. Describe the algorithm clearly. Write clean, clear, modular code. The most important part is the actual code for comparing, not input/output code or low-level data access code.

If you see ambiguities or inconsistencies with the above rules, describe them, resolve them, and code accordingly.