I used a brute force strategy and algebraic methods to reduce complexity.

Pseudocode

Declare needed variables

Output a prompt

Take in input save to variable

Switch(input)

Case (a):

Declare variables

Loop a

Count++

Loop e

Count++

Solve for d

Count++

Solve for f

Count++

Check validity of f

Loop b

Count++

solve for c

Count++

Verify constraint equations with if statements

Print variables and their values

Print nva count

Exit program

Break;

Case(b):

Declare variables

Loop a

Count++

Loop e

Count++

Solve for d

Count++

Solve for f

Count++

Check validity of f

Solve for g

Count++

Solve for j

Count++

Check validity of g

Loop b

Count++

solve for c

Count++

Loop i

Count++

Solve for h

Count++

Verify constraint equations with if statements

Print variables and their values

Print nva count

Exit program

Break;

Case(c):

Declare variables

Loop a

Count++

Loop e

Count++

Solve for d

Count++

Solve for f

Count++

Check validity of f

Solve for g

Count++

Solve for j

Count++

Check validity of g

Loop b

Count++

solve for c

Count++

Loop i

Count++

Solve for h

Count++

Solve for o

Count++

Loop m

Count++

Solve for k

Count++

Solve for n

Count++

Loop l

Count++

Verify constraint equations with if statements

Print variables and their values

Print nva count

Exit program

Break;

Default:

Print error statement

Return 0;

First ask for input then use it to go into a switch statement. Within the cases start looping variables needed to be able to solve some equations. Solve those equations keep looping if needed. Once all variables are being used, use if statements to check constraints. If solution is found print. Repeat for next two cases.

I didn’t really do a pre-analysis, instead I solved equations for variables which helped reduce search complexity. This helped reduce the number of for loops used as opposed to my original path of solving the problem.

My code wasn’t generic enough to use for other constraint problems, however my program can be extensible by adding onto the current constraints.