1. Each team member must provide a separate solution to the problem.  You may use Java, C, C++, or C# for your solution
2. There must be an additional team solution, that is built based on evaluating each member's solutions.  This is the 'official' solution for your team.
3. Establish coding standards (naming conventions, whitespace use, placement of { }, comment standards, modularity considerations, etc.).  All solutions to the problem must follow these established standards.  Include a .pdf of the standards you established. Here is an example of what one team produced last year: [General Coding Standards.pdfPreview the documentView in a new window](https://canvas.ewu.edu/courses/921712/files/28602017/download?wrap=1)
4. The input from the specified problem will come from console input.  The output is also to the console.
5. As a team, build a robust test file that includes the sample input given in the problem statement as well as tests for each piece.  Note that your test file need only contain valid data as specified in the problem statement.  More specifically, you do not have to account for malformed data as part of your testing.
6. Solve this problem individually, then as team WITHOUT SEARCHING FOR SOLUTIONS TO THIS PROBLEM.  This problem has been solved many times.  Tom has those solutions.  Part of this exercise is for you to practice your problem solving skills.  If you get stuck, seek help from a team member or Tom!

To turn in (team lead will submit)

* Submit a zip file that contains
  + (50 points) Source code for all solutions (individual team members and official team solution).  Each solution should be named such that it is clear whose solution it is.
  + (25 points) .pdf of your team's coding standards -- put serious effort into this as it is very important
  + (15 points) The robust input file you built to test your solution(s)
  + (10 points) Sufficient output captures (these can be screen shots or redirected output files) to demonstrate your solution operates correctly