Collaborative Programming Exercise

In today’s activity, you’ll be writing a complete program with another programmer. It is crucial that you clearly outline the purpose of code using comments so that your partner programmer will understand the purpose of what you’ve already composed. You may not work with your partner programmer on your section of code, and all communication between partners must occur via comments included in your program.

As Programmer 1 on this project, it is your responsibility to **start** writing a program called ProcessingNumbers that does the following:

1. Accepts as user input a string of numbers.
2. Prints the smallest and largest of all the numbers supplied by the user.
3. Print the sum of all the even numbers the user typed, along with the largest even number typed.

Programmer 2 will complete the program you started, so it is important that you leave clear directions and comments in your code. You will be the last programmer to work on the project, so you should look for Programmer 1’s notes and edits so you can understand which edits you want to keep, or which edits you need to reverse and why.

Begin by diagramming the design and structure of your program here below. The more time you spend on this, the more likely your partner programmer will be able to finish the program as you wanted it. This section is required.

Programmer 1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Programmer 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

You should begin writing your code in the space below. Include comments to let Programmer 2 know what each segment of code is supposed to accomplish, and write all of your code in pen. If you want to correct something, cross out with a single line so your partner programmer can see what used to be there.

Reflection & Feedback

Programmer 1, what did you notice that worked smoothly during this exercise? Did Programmer 2 complete your program the way you thought they would? What was did they do that was unexpected? Did you find their edits and feedback clear and easy to understand? If not, what would you recommend they change in the future?