|  |  |
| --- | --- |
| Drive Full Name | Jose Carrillo |
| Partner Full Name | Lucas Podowski |
| Student ID | 1867377 |

Reflection:

This assignment posed several challenges, especially in committing and pushing the flowchart to GitHub and simplifying the algorithm to avoid overly specific code references. One of the primary difficulties was ensuring that the flowchart was clear and easy to follow, even though the process involved several decision-making steps and calculations. Translating a logical problem like this into a visual representation was an exercise in balancing clarity and accuracy while avoiding excessive complexity.

Committing the flowchart to GitHub was a learning curve, as ensuring that it met both the visual and organizational standards required some extra steps. I encountered issues during the push and commit phase, especially when handling conflicts in Git, but these were ultimately resolved by referring to resources and debugging patiently. Keeping track of the flowchart on both local and remote repositories while managing its updated versions taught me the importance of methodical version control. This also reinforced the necessity of keeping the code base clean, organized, and thoroughly commented for future reference and collaboration.

When writing the algorithm, one challenge was simplifying the code without making it too abstract. Using less specific code wording while preserving the logic required careful reworking of some sections, especially around the conditional logic with If and Else statements. These challenges were overcome by constantly testing and ensuring that each section of the code executed as expected. Once the code was cleaned up, I successfully submitted the flowchart and algorithm through Moodle, confident in the simplicity and clarity of the final product.

In terms of key takeaways, I realized that if and else statements are often more efficient than constantly referencing cells when using Excel for calculations. This made the process smoother and more organized, reducing the chances of errors. Simplifying the rows and keeping them clean was a valuable lesson that reinforced the benefits of code clarity. This reflection also underscored the importance of keeping decision-making steps as simple as possible while ensuring functionality remains intact.

This lab reinforced many key concepts I thought I already understood but helped me gain a deeper, more practical understanding of the programming process. It improved my confidence in writing efficient code and organizing my thought process for both algorithms and flowcharts.

Working with my partner Lucas was an added benefit, as he had a solid understanding of the algorithm and was able to write the code quickly. This efficiency in our partnership allowed us to spend more time on debugging and fine-tuning the flowchart, ensuring that the final submission was polished and ready for review.