Assignment 3a Group Part

Byron Kooima, Matthew Morse, Jonathan Raleigh, and Alvaro Salinas

CS/161

January 29, 2017

Tim Alcon

Which project plan did your group decide on?

The group decided on the project plan submitted by Matt Morse.

What advantages do you think Matt's project plan has over the others?

One strong advantage that Matt's project plan has over the others is the readability, organization, and logical correctness of his flowchart.

When it comes to readability, we concluded that Matt's flowchart model is superior to pseudocode given that the former would likely be more useful in a team setting. Pseudocode is useful for quickly pulling together the steps required to generate functioning code, but with a flowchart you have the added advantage of easily sharing it with other people. This readability can definitely come in handy when working on projects with a team, which in a professional setting often includes individuals with little to no programming experience..

In terms of organization, Matt's flowchart was the most tidy and visually intuitive. The natural progression of the steps helps the reader quickly attain a general understanding of the project, and the clear descriptions for each step also provide more in-depth understanding when needed. This makes it a great tool for other team members as well as a reliable reference while writing the code for the programmer building the project.

In terms of logical correctness, Matt's flowchart had no errors. This is a clear advantage to other designs that might have some logical flaws, given that it can save the programmer a lot of time when writing out the code.

Another advantage that Matt's project plan had over the others was that it had a much more robust and comprehensive testing plan, with 11 different scenarios to test. Each test plan was unique and also provided the best tests for extreme and benign inputs. All major potential points of trouble were covered in the plan (such as only entering zero, entering very big and very small values, entering multiple same values, entering positive, negative and zero, etc). And upon further review of the assignments specs, his testing plan adhered the most consistently to the desired format of two columns with test descriptions in the first column and expected output in the second.

What improvements do you think could be made to that project plan?

While the group was largely in agreement over the soundness of the testing plan and design, there were multiple suggestions for improvement that were primarily in regard to formatting.

Looking at the testing plan specifically, it was rightly pointed out that the testing and design documents were in reverse order within the PDF files. This is important since technically the test plan should be completed before the pseudocode or flowchart. Some of the group's other testing plans made use of different layouts and descriptions that, although they did not exactly follow the formatting in the assignment's guidelines, offered definite benefits. Byron, for example, preceded his testing plan with brief reasons behind each test. With a more complex program, this method would certainly prove useful for spotting logic errors if the program writer and tester were two different people. Furthermore, Alvaro utilized three columns in his testing plan rather than two. Again, especially with a more complex program, this layout would likely cut down on errors by more clearly delineating the processes involved in carrying out a testing plan.

As for the program design, the consensus was that the logic was sufficient to produce a working program. There were, however, a few solid points for improvement. Though it largely applies to this specific problem, initializing the min/max values before the loop would simplify the code and ensure that the variables will apply to the entire scope of the main function. With reference to readability, adding more detail to the wording of the flowchart's steps would make it easier for someone who was unfamiliar with project to understand--so as long as the amendments adhered as much as possible to natural language. For example, "get number of integers from user" could be changed to "get number of integers that user wishes to enter." One final alteration would be to italicize variables wherever they are referenced for clarity and emphasis in the flowchart.