Sara Hrnciar
450 Independent Study
Dr. Gourd
Program Profiler

I initially sought to create a program which determines the likelihood of a student's work being influenced by an outside source such as chatGPT. This was an incredibly steep goal to achieve in only several months. Ultimately, I ended up with a program which takes a folder of student's previously written programs and develops a "program profile," for the student. This is done by converting each program to an nxm matrix, where n is the number of lines in the current file and n is the number of predefined programming keywords. After converting each program to a matrix, they are each interpolated into x rows, where x is defined by the number of lines of the folder's longest program. This is done by adding new rows in between values which fit well to the already existing matrix. From here, each matrix's contents is scaled by 1/p, where p is the number of programs factored into the profile. This is done to ensure the program matrix's contents are sized proportionally to a single program matrix. The summed weighted matrices then give a "program profile," or a matrix which represents a student's typical program structure. From here, students' future programs are then compared to this program profile. The user is left with a similarity score from 0 and above. A higher score signals a lower similarity.

I think the largest issue I've had has been figuring out the best way to approach the program profile. From my chosen interpolation technique, to weighing the matrices, to normalizing the program profile/current program, to comparing their normalized values, I'm still not sure if my methods provide the most accurate results. For example, there were various ways to interpolate the matrices and I chose the one which I personally felt would fit best (as described previously), but there could have been another more well-fit one. I also don't know the best norm to take of the matrices, and I don't know the best way to create the similarity score. It seems that what I have right now works, but I'm sure I could use different methods to achieve more accurate results. Taking linear algebra alongside this independent study was helpful, but I think I have much more to learn in order to make this tester as accurate as possible.

I also don't necessarily know how to interpret the result, as the similarity ranges from 0 and above. My result 0 came from comparing the exact same programs, and 12 came from comparing a program with a text file containing "panini\npanini." The way this score is determined is through finding the Euclidean norm of both the student's recent program and their program profile, and taking the absolute value of their differences. Ultimately, this measures the matrice's "distances" from each other. An issue arises when a user is met with a similarity score of, say, 5. What does this mean? Does it hint at help from an outside source, a minor change in a student's typical program structure, or does this mean that the newest program's structure is totally different? I know that the program score adjusts accordingly in obvious tests. The program returns a score of nearly 0 given the program profiler folder contains a text file identical to the currently tested program. It returns a similarity score of 12 when the folder contains a textfile of random words. The similarity score goes closer to 0 as you add a higher quantity of student programs (which are identical to the currently tested file) to the profiler program. Ultimately, though, the resulting number does not hold a concrete value to it and leaves interpretation up to the user.

There is a lot more that I can add to develop my program profiler. The first thing is to create a program profile for a collection of programs from chatGPT and to also compare student's programs to that. I also could incorporate something along the lines of an accuracy check, which could output how similar the programs in the program profile are. That way, if a program profile is all over the place and each program considered is vastly different from the others, it could be taken with a grain of salt. On the other hand, it would be much more suspicious if a highly accurate program profile has a high difference from a student's recent program. I could also develop a gui and make it so that the user can drag and drop files. Ultimately, there are endless considerations I can factor in to increase the accuracy of the program, and there are a lot of additional applications that can be added to this program.