

## Banker's Algorithm Documentation

I solved this problem using the language C++ and the IDE Microsoft Visual Studio. After completion, I copied my source code into a test file on the wasp/hornet Kent State servers to ensure they also worked in that environment.

This program requires that a text file is given as a command line argument. I included the text file that is used that is in a working format.

I defined four functions, `printMatrix()`, `getProcess()`, `rowIsLess()`, `getNewAvailable()`. `printMatrix()` prints a full matrix passed, I used this function for readability.

`getProcess()` returns one desired row from a matrix and used for comparison.

`rowIsLess()` does the actual comparison mentioned above.

`getNewAvailable()` calculates the newAvailable value on a true evaluation within the `main()`.

I used multidimensional arrays to represent matrices and I used vectors to represent a single row from a matrix.

I used a queue to keep track of processes that failed evaluation within the `main()`. This way the front item was also the first failed.

My main loop uses a while loop and compares an integer, initialized to zero to the number of allowed processes. After this loop ends there is a second while loop that checks for the queue of failed processes being empty, or a safety mechanism, evaluating to true. I used an integer, also initialized to zero and make sure it also always less than the number of processes for the safety mechanism.

There are a large number of comments within my program for further explanation if needed.