Assignment 1 Test Discussion and Design Choices

Name: Jeremy Trendoff

Student Number: 101160306

Test Discussion

Test Case 1, The Mandatory Test Data.

```
The input matrix used:
{20, 20, 50}
{10, 6, 70}
{40, 3, 2}

The resulting determinant of the matrix is: 41140
The largest number in the matrix is: 70
Elapsed Time: 240 micro sec
```

When checking the results of this test, I found that the determinant and the largest number was correct.

Test Case 2

```
The input matrix used:
{20, 10, 30}
{100, 6, 2}
{40, 5, 2}

The resulting determinant of the matrix is: 6640
The largest number in the matrix is: 100
Elapsed Time: 190 micro sec
```

When checking the results of this test, I found that the determinant and the largest number was correct.

Test Case 3

```
The input matrix used:
{0, -20, 500}
{1, 60, 7}
{3, 30, 2}

The resulting determinant of the matrix is: -75380
The largest number in the matrix is: 500
Elapsed Time: 279 micro sec
```

When checking the results of this test, I found that the determinant and the largest number was correct.

Test Case 4

```
The input matrix used:
{0, 0, 150}
{15, 4, 70}
{44, 36, 21}

The resulting determinant of the matrix is: 54600
The largest number in the matrix is: 150
Elapsed Time: 239 micro sec
```

When checking the results of this test, I found that the determinant and the largest number was correct.

Design Decisions

For this assignment, I used a shared memory state containing an array of length 3 to hold the largest numbers in each row, and an integer determinant to calculate the resulting determinant of the matrix. This struct was declared in the SharedMem header file.

I wanted to keep my main method as readable as possible. To do this, I delegated large chunks of code to their own functions. The functions created were get_largest_number_in_matrix_row(), get_largest_number() and print_matrix(). get_largest_number_in_matrix_row() is used by each child process to find the largest number in their respective row. get_largest_number() is used to find the largest overall number in the matrix, and print_matrix() prints the test matrix used to the console. If these methods were not implemented, there would be a lot of repeating code which would make the main function seem cluttered.

Another design decision I made was to not have a separate method to calculate the determinant. I did this because it would require determining which process is running the function from within the function. Plus, the code required to calculate the determinant was only one line and I did not feel that it compromised the readability of the program at all. This resulting in a shorter program as well.

Overall, my design choices were made specifically to make the main function as readable as possible, and without these changes, there would be lots of repeating code causing main() to be overly complicated.