

Program 2 Test  
CSI4337

Compile with options: `-Wall -ansi -pedantic -g -std=gnu99 -lpthread -lm`  
Run with: `valgrind --tool=memcheck --track-fds=yes --show-reachable=yes ./<executable>`

Part I (50 points)

0 points if program fails to compile.

If program fails to operate properly, deduct 50% and grade by inspection.

Verify correct use of `pthread_create()` and `pthread_join()`. In particular, the argument passing and return value recovery. They are not required to actually use the return value slot. Any reliable technique for communicating with the threads should be fine.

Verify each thread has unique random number sequence (i.e., unique seed per thread)

They are not expected to include synchronization code (other than the join). This assignment should work just fine without it. Do not deduct if they included synchronization, but point out improper use of synchronization.

Test with 1, 4, and 6 threads.

Part II (50 points)

0 points if program fails to compile.

If program fails to operate properly, deduct 50% and grade by inspection.

For both parts:

Take off points for poor coding practices

- No unjustified global variables, etc.
- Verify that system functions, etc. check for errors.
- Declare variables with use (not old C-style predeclaration) – Just mark this; don't take points off

MARK            unused include or unjustified include

-5            for incorrect behavior or no/poor usage message on 0 and 2+ parameters

-10           for poor commenting

-X            for compiler warnings

-2            for each unjustified numeric constant

-5            bad submission

-1            for each section of commented code

-5            if violate coding convention