Assignment 4 – Scheduler Showdown

Important notes:

- **Your code must run properly on the school's Linux machines using the GNU compiler (g++ / fg++).**
- Using a language other than C++ will be -10%, if you use a different language you must include a makefile.
- Non-compiling code will receive a grade of 0%.
- Segmentation Faults, Core Dumps, and other fatal runtime errors will receive a grade of 10%.
- We expect that your code is consistent and readable.
  - Use the same formatting throughout your code for variable names, functions, and classes.
  - Make sure to have clear and productive documentation, confusing code should be commented to be readable.
  - comments should provide something not "//this is an int".
- You may expect that files will only be used as command line arguments if they exist and are formatted properly.
- Make sure all output is sent to std out.

	Perfect	**Good**	**Bad** (0 points)
Code	Compiles **(10 points)**		Doesn't compile
	Correctly implement Shortest Process Next algorithm **(25 points)**	Works for some but not all test cases or implemented with partial correctness **(pts/test)**	Not the correct algorithm or does not work
	Correctly implement Shortest Remaining Time algorithm **(25 pts)**	Works for some but not all test cases or implemented with partial correctness **(pts/test)**	Not the correct algorithm or does not work
	Correctly implement Highest Response Ratio Next algorithm **(25 pts)**	Works for some but not all test cases or implemented with partial correctness **(pts/test)**	Not the correct algorithm or does not work
Statistics	Calculate and output: Finish time, Normalized turnaround Time, Avg. Turnaround, and Avg. Normalized Turnaround **(3pts each)**	Incorrect calculations, bad output formatting (does not follow sample and is hard to read / understand), not outputting all statistics	Did not calculate statistics