Paul Vorobyev (pv149) and Rui Zhang (rz187)
Systems Programming
Section 06
Assignment 2 - Pointer Sorter

Our programs, outR and outT, use processes and threads respectively to employ a "Length of Long Sequence" (LOLS) algorithm in order to compress files. The code is separated into several files:

- LOLS.c/LOLS.h contain the compression function that takes in a char * array and performs the LOLS algorithm
- parts_compressor.c/parts_compressor.h contain contains functions and structs that are shared between outR and outT
- compressR_LOLS.c/compressR_LOLS.h uses processes (via fork()) to call compressR_worker_LOLS
- compressR_worker_LOLS.c/compressR_worker_LOLS.h is the worker component of the multiprocess implementation of LOLS
- compressT_LOLS.c/compressT_LOLS.h is the multithreaded implementation of LOLS

The way the LOLS algorithm starts by allocating an array equal to the input size. It goes through this array, comparing a target with every element of every character until a discrepancy is found. Once this discrepancy is found, the target character and total count are written to a block. This is not done if the count is 1 or 2, since additional space would not be saved. This algorithm will run through the input array only one time, so the running time complexity of LOLS is `O(n)`.

To simplify the compilation process, a Makefile has been provided. To compile the process implementation of LOLS, use `make outR`. To compile the multithreaded implementation of LOLS, use `make outT`.