Lab 06 exercise 02

Write a C program that takes as arguments a number C and a directory name dir. The main program, using the system call system, outputs in a file `list.txt` the list of files in directory dir. Then it forks C children, and loops

- waiting on a pipe `request_pipe` that a child asks the filename of the file that it will sort,
- reading the next filename from file `list.txt`, and passing it the queue represented by pipe `data_pipe`
- waiting on another pipe `answer_pipe` the child identifier, and the number of lines that the child has sorted

until all the files have been sorted.

Finally, it prints the total number of files sorted, the total number sum of the lines that have been sorted, and the sum of the lines that each child has sorted.

Each child process loops

- sending a request (just a byte) to the parent process through `request_pipe`
- waiting on `data_pipe` the filename passed by the parent
- sorting the file by means of a system call `system`
- sending through `answer_pipe` its identifier, and the number of lines that it has sorted

until it receives a signal `SIGPIPE`, which inform it that the parent process has closed the read terminal of `request_pipe`.

After all files listed in `list.txt` have been sorted, the main process must produce a single file `all_sorted.txt`, where all the numbers appearing in all the sorted files are sorted in ascending order. Do this by using again system call `system` with the appropriate `sort -m` command.