University of Toronto
Department of Computer Science
CSC 228 - Summer 1993

Assignment 1: Prison Management

Due: in tutorial, June 1

This assignment will give you some practice with separate compilation, file I/O, good modular programming style, and a better appreciation of the job of a prison warden.

You are to write a program, make up test data for all the necessary files, and write a report. The report is intended to convince your tutor that your program is correct. Read pages 1-2 of the *course guide* for more information.

The problem

Shrinking government budgets have made it necessary to close several Ontario prisons and transfer the inmates to a larger facility, recently constructed outside of Toronto. In order to effectively manage this new prison, the administration requires two lists of inmates, one sorted in alphabetical order by prisoner name (last name first), and the second sorted by prisoner identification number. Coincidentally, prisoner identification numbers follow the same format as University of Toronto student numbers. Inmate lists from the smaller prisons are available, but these are not in sorted order.

Your task – Part I

Your first task is to write a program which inputs prisoner records from character-based files belonging to the smaller prisons, and outputs two sorted character-based files as described above. You may use any sorting method you please, but the source of your code must be acknowledged. Name this program "sort.c."

After the input files have been sorted, you must combine or *merge* the sorted files into two large files for the new prison facility. The file sorted by prisoner name should be called "inmate.names" and the file sorted by prisoner identification number should be called "inmate.numbers." You may assume that n, the number of smaller prisons which are being closed, is a power of two, although you will receive bonus marks if your program works elegantly for arbitrary n, and is well-described in your report. Name this program "merge.c."

The main program which calls functions in "sort.c" and "merge.c" should be named "prison.c." If you wish to include a number of constants or data structure definitions which are shared by two or more of the program files in a common header file, then name this file "prison.h." You should have no need for global variables for this programming assignment.

Your task – Part II

Impressive as the new prison facilities may be, they do not include provisions for serious offenders. Therefore, instead of being transferred to the new prison, these criminals will be placed in high offices of government. A single file containing a list of all such inmates from all the smaller prisons, sorted by prisoner name, is available.

Your second task is to make use of the list of serious offenders by excluding these from the "inmate.names" file for the new facility during the merging step. Name this program "exmerge.c." For this component of the assignment, you do not need to provide an output file sorted by prisoner identification number.

Developing your program

You will find life quite simple if you first build a program which just sorts the input files. Test this program thoroughly before continuing. Then, write the merge routine, ignoring the additional requirements of part II. Again, test this program thoroughly, with the sample files you have created, and gather a small but sufficient number of test outputs to demonstrate to your TA that your program works.

Once this is done, you should copy "merge.c" to "exmerge.c" and make the small changes necessary to complete part II of the assignment. *If you have written your program correctly, it should only be necessary to change that one file.*

Your report must discuss both versions of your program, and you must submit both versions as appendices to the report. Do not forget to include your test input and output for both versions, and label them well.