

The format you want to use for messages in the files:

Since no modern data structures are allowed, the data within the files would have to be read and accessed as least as possible. Thus, there would need to be either spaces or markers within the text files to separate messages and other information accordingly, and have the program recognize these markers to separate the messages. `find_print_msg(1)` for example could not just get the index of 1, but what could happen is that it can loop through the message strings within a file and match according to the number of spaces/markers/etc to get the details of said message.

The relationship between messages and files:

Messages would be strings/integer combinations that would lie within the files and be accessed by the program at a given time, especially when using a method such as `find_print_msg(1)` which directly needs access to the messages.

Any files you want to create other than ones that hold message contents:

There might be files needed purely for data arrangement, especially when needing to access data. Files could be created to store indexing information a hashmap or dictionary would, or there could be files whose purpose is simply to hold data following a reset in the best way so that it can be saved and able to be accessed again as efficiently as before.

A brief justification for your choices:

All of these choices are to abide by the set limits and ensure that time complexity is met when trying to search for any bit of information. Having there be denotations within files rather than spreading information too thin across many files would save a lot of time in searching, keep data organized across many restarts, and perhaps find ways to store more than possible when it comes to the number of people/max number of files allowed in rotation.