

Divide & Conquer

Create a C++ program to ingest, sort, and display a menagerie. Each animal in the collection will have a name, type, and age. Names and types are text and ages are integers. Your program should input the list of animals from a text file that is whitespace delimited (see the example below). Use a `std::vector` to store the menagerie.

Display a menu for the user with the following options:

1. Quicksort list by name
2. Quicksort list by type
3. Quicksort list by age
4. Mergesort by name
5. Quit program

Menu items 1-4 will perform the action specified and then display the sorted list. Use a range-based for loop and `std::print` (or `std::println`) to display the list.

Your program should be decomposed into small, single purpose functions (or appropriate member functions). You may use a structured or OOP approach for your design. You may use `std::swap` or `std::iter_swap` in your sorting algorithm implementations. You may not use the built-in or library sort routines for this assignment. Implement quicksort and mergesort from the algorithms we discussed in class.

► Do not use `std::vector` operator `[]`. Use the `std::vector` at member function to access elements of the vector. See the STL documentation for why `at` is a better choice.

Your program must compile without any compiler warnings or errors using the c++ compiler on the CS server with the C++23 standard flag. Your program must exhibit an appropriate level of decomposition, good structure and organization, produce correct results, be efficient, and have a consistent layout and style.

Example animals in input file:

```
Yogi Bear 6
Brian Dog 4
Muttley Dog 6
Bigelow Mouse 3
Hoppy Hoparoo 2
Sancho Panda 5
Mumbly Dog 3
Atom Ant 1
Bugs Bunny 7
Axl Gator 2
Pinky Mouse 1
```

Use a `constexpr` to store the file name (“use named constants for literals”).

Here is an example C++ style best practices guide for code layout:

<https://lefticus.gitbooks.io/cpp-best-practices/content/03-Style.html>.

You may use any reasonable C++ style (ask if you’re not sure), but the #1 rule is to be consistent.

<< Continued on next page >>

Due dates:

Monday 09/21 before 11:59 p.m.

How to submit:

Create a folder named “assignment-2” on the CS server. Put all your submission files in this folder. A Makefile in your folder would be appreciated.

Program evaluation:

Your program must compile without any compiler warnings or errors using the ++23 standard. Code will be evaluated for an appropriate level of decomposition, structure and organization, correctness, efficiency, and style.