

1 Selection-Sort them all

In this exercise we will combine the advantages of generic programming and “modern” C++ iterators.

The goal is to read an input file and output its content in a sorted manner. The difficult part is that the type of the elements stored in the input files may vary from one to another. To clarify this, the first line of the file contains a type identifier. It is one of the following: i, u, f, d, c, str for int, unsigned int, float, double, char or string.

Secondly, the file also specifies which standard container is to be used as a buffer. The second line is one of the following, referring directly to the container defined in the standard: `std::list`, `std::vector`, `std::deque`.

You must then read the file into buffer, sort it and write the sorted elements, prefixed by element type and container, into a new file.

To make things even more interesting, you must code your own sorting algorithm with the following restrictions:

- The sorting method is selection sort
- Your algorithm must use iterators
- There is only one (generic) implementation of your sorting algorithm