
C++ PROGRAMMING TASK: ADDRESS BOOK¹

SUMMARY

Using standard C++ and only its standard library, design and implement code that models an address book.

MINIMUM REQUIREMENTS

- Support from zero to a “large number” of entries (typically 100,000s).
- Entries should contain:
 - A first and/or last name (you can treat each character as a single byte, i.e. no need to support Unicode).
 - An optional phone number.
- Provide functionality to:
 - Add an entry.
 - Remove an entry.
 - Retrieve entries in alphabetical order, sorted by first name.
 - Retrieve entries in alphabetical order, sorted by last name.
 - Retrieve entries whose first or last name exactly or partially match a supplied search string, from the beginning of the name (e.g. searching for “Dan” would match any entries with “Daniel” as well as any “Dan”).
 - Make this matching case insensitive.
- Write a short main() function that demonstrates all of the address book’s functionalities:
 - It is fine to hard-code example entries.
 - It is fine to write to the standard output.
- Provide documentation for the API and to justify your design choices.
- Use only features up to and including C++14.
- Provide instructions on how to build your code.

ADDITIONAL CONSIDERATIONS

- Imagine that the address book code could be used by other developers, in various client applications.
- Assume that the retrieval operations are going to be called the most frequently in the majority of client applications.

NOTES

- You will be given up to three days to think about and implement your solution, but this is not intended to be a big programming task. There is no limit on the amount of time you should spend on it but bear in mind that most candidates take around two to three hours to complete. When submitting your solution, please let us know approximately how much time you spent.
- Feel free to use C++ and standard library references, e.g. cppreference.com, but please do not ask for help with the specifics of this task. It should be your own work.
- Feel free to comment on any known shortcomings of your design and make suggestions about how these could be addressed given more time or if the library and version restrictions were removed.

¹ Version m.3 (29/10/2020)