QA DOCUMENTATION - OLOUXII

# 1. Overview

The goal of this QA documentation is to describe the quality assurance process for the application OlouXii – a program for storing, searching, and processing historical events using a linked list.

# 2. Scope of Testing

The testing covers:

- Adding events (at the beginning, at the end, and in sorted order)

- Editing and deleting events

- Saving to and loading from a file

- Searching by name, date, year, and historical figure

- Sorting by date

- Visualization of the list

# 3. Testing Strategy

## 3.1 Types of Testing

- Unit testing – each function is tested individually (e.g., addFirst, deleteEvent, searchByName)

- Integration testing – checks how functions work together (e.g., create -> save -> load)

- Manual testing – via the CLI menu

- Boundary testing – e.g., entering invalid dates

## 3.2 Tools

- Compiler: g++

- Manual input through the terminal

- Text editor: VS (Visual Studio)

# 4. Test Cases

|  |  |  |  |
| --- | --- | --- | --- |
| # | Description | Steps | Expected Result |
| 1 | Add event (addFirst) | Enter valid data | Event appears at the beginning of the list |
| 2 | Add with sorting | Select "Add new event and resort" | Event is inserted based on date |
| 3 | Save to file | Select Exit | `events.txt` contains the data |
| 4 | Load from file | Restart the program | Data is displayed automatically |
| 5 | Search by name | Enter an existing name | Correct event is displayed |
| 6 | Search by date | Enter an existing date | Correct event is displayed |
| 7 | Delete event | Enter name | Event is removed from the list |
| 8 | Invalid date | Enter 31/2/2023 | Normalized to 3.3.2023 |
| 9 | Empty list | Select "Show full list" without events | Shows empty output |
| 10 | Re-adding duplicates | Add several identical events | All are saved without crash |

# 5. Success Criteria

- No runtime errors

- Events are saved/loaded correctly

- Each search operation returns the correct result or appropriate error message

# 6. Recommendations for Improvements

- Input validation (e.g., for empty name)

- Add unit tests using Catch2 or Google Test

- GUI interface for easier testing

- Undo/Redo functionality