Programming Project / C++

– To-Do List App –

Student 1: Tanas Andreas Dorian

Student 2: Mihalache Madalina-Andreea

I. Task Description

- Student 1 – List Manager Responsibilities:

• Create new to-do lists with titles
• Add, modify, and remove tasks from a list
• Save and load lists from persistent storage (todo_lists.txt)
• View available lists and their contents

- Student 2 – Task Processor Responsibilities:

• Load an existing to-do list
• Mark tasks as completed
• View only active (incomplete) tasks
• Submit finished lists (delete or archive them)
• Log completed actions to a file (task_log.txt)

II. Data Structures Used

The following C++ classes/structs will be used:

- Task:

• std::string description
• bool completed

- ToDoList:

• std::string title
• std::vector<Task> tasks

- std::vector<ToDoList>: Maintains collection of lists (App 1)

- std::map<std::string, ToDoList>: Used in App 2 to process one list at a time

III. File Structure

todo_lists.txt: Stores the master list of all to-do lists

```
-= List Title =-
[ ] Task 1
[x] Task 2
...
```

task_log.txt: Maintains logs of completed tasks and processed lists by App 2

```
[2025-05-20] List: "Daily Tasks"
✓ Clean kitchen
✓ Finish report
```

IV. Interacting with Executables

- Application 1 (List Manager):

```
./list_manager.exe view_lists
./list_manager.exe create_list <title>
./list_manager.exe add_task <title> <description>
./list_manager.exe delete_task <title> <task_index>
./list_manager.exe modify_task <title> <task_index> <new_description>
```

- Application 2 (Task Processor):

```
./task_processor.exe view_tasks <title>
./task_processor.exe complete_task <title> <task_index>
./task_processor.exe submit_list <title>
./task_processor.exe log_status
```