Programming Project / C++
-= Incident Reporting System =-

Student 1: Szekelyhidi Deborah
Student 2: Stefanov Carla-Bianca

I. Description

Student 1 is responsible for managing the incident reports:
- Add, delete, update incidents (type, description, location, date)
- View all submitted reports

Student 2 is responsible for interacting with the system as a user:
- Submitting a new incident report
- Viewing/filtering existing reports by area or type

II. Data Structures Used by the Team

We will use the following classes:
- Incident: string id, string area, string type, string description, Date date
- Date: int day, int month, int year

III. File Structure

We will use the following files:

- incidents.txt

File to store all reported incidents, structured as:
<number_of_incidents>
<id> <area> <type> <description> <day> <month> <year>
…

- user_reports.txt

Temporary file where user-submitted reports are saved before being added to the main incident list:
<area> <type> <description>

IV. Interaction with Executables

Application 1 (Admin Operations)

app_admin.exe provides the following functionalities:

- ./app_admin.exe view_all_incidents

View all reported incidents.

- ./app_admin.exe add_incident <id> <area> <type> <description> <day> <month> <year>

Add a new incident to the system.

- ./app_admin.exe delete_incident <id>

Delete an incident by its unique ID.

- ./app_admin.exe update_incident <id> <field_to_update> <new_value>

Update a specific field (area/type/description/date) of an incident.

Application 2 (User Interface)

app_user.exe provides the following functionalities:

- ./app_user.exe report_incident <area> <type> <description>

Report a new incident. The system will automatically assign the current date.

- ./app_user.exe view_reports

View all incident reports.

- ./app_user.exe filter_by_area <area>

View reports filtered by area.

- ./app_user.exe filter_by_type <type>

View reports filtered by type.