User Manual – Water Distribution Simulation System

1. Starting the Program

1. Make sure all source files are compiled using the provided Makefile

Run the following commands in terminal:

```
make
(To compile all files)
./graph_app
(To run the application)
make clean
(To delete all object files)
```

2. Once you run the executable, the program initializes the water network with:

• A reservoir (source tank)
• Multiple tanks connected through pipes

3. Random initial water levels and connections are displayed.

2. How the Simulation Works

The system runs in time steps (usually every 30 seconds). Each step:

1. Reduces tank water levels (simulating usage).
2. Identifies tanks below the prescribed level.
3. Refills them from the reservoir.
4. Prints supply and delivery details.
5. Warns if there is a possible leak.

3. Typical Output Format

Example of output after each interval:

```
--- Priority-based refill sequence at 00:30 ---
Processing Tank 2 (Tank B) - Priority: 250 - Level: 150 / 500
Expected delivered (units): 120 | Actual delivered: 118
--- Snapshot at 00:30 ---
Node 0 (Reservoir): level=inf / inf
Node 1 (Tank A): level=480 / 500 | valveStatus=1 | outgoing=1
Node 2 (Tank B): level=270 / 500 | valveStatus=1 | outgoing=2
Edges:
Edge[0] 0->1 cap=100 flowRate=100 active=Y valve=1
Edge[1] 1->2 cap=80 flowRate=80 active=Y valve=1
```

4. Leak Detection and Alerts

If a leak is detected, the program displays:

```
>>> Leak suspected on path to Tank 2 (expected=120, actual=60)
Alternate route found. Attempting alternate route...
Alternate route delivered adequately.
```

If no alternate route exists, it shows:

```
No alternate route available. Please inspect pipes or mark leak repaired.
```

5. User Interaction and Editing Options

After each simulation step, the user may be prompted with:

```
Leak detected on pipe 1->2. Has it been repaired? (y/n):
```

Typing 'y' repairs the pipe; 'n' keeps it broken.

```
Would you like to edit any pipe or tank properties? (y/n):
```

If yes, you can edit tank capacity, flow rate, edge status, or valve settings.

6. Simulation Log Access

You can view recent log entries:

```
Do you want to see last K log entries? Enter K:
```

Example output:

```
[00:30] Tank 2 consumed 5.2 units
[01:00] Leak suspected on path to Tank 2
[01:30] Alternate route delivered adequately.
```

7. Ending the Simulation

The simulation runs continuously until manually stopped with Ctrl+C. It then displays the final tank levels and pipe statuses.

8. Quick Summary of Choices

Prompt or Option	What It Does
y / n for leak repair	Marks pipe as repaired or keeps it broken
Edit node/edge	Change tank/pipe details like capacity or valve
View last K logs	Shows recent activity in reverse order
Automatic updates	Runs every 30s; prints tank levels and refills

9. Expected Outputs

Action	Printed Information
Tank usage	Tank X consumed Y units
Refill	Supplied to Tank X expected=Y actual=Z
Leak detected	>>> Leak suspected on path to Tank X
Alternate route	Alternate route delivered adequately
Editing	Edge 1->2 capacity set to 100
Snapshot	Shows tank and pipe status