CSCI.603.2

Subject	Advance C++ and Program Design
Author	Shivkumar Dudhani, Krish Godiawala
Project Description	A generic solver to solve puzzles based on Breath First Search
Files	Configuration.h, Puzzle.h, Clock.h, Clock.cpp, Water.h, Water.cpp, Solver.h, Solver.cpp
Platform	C++ 11

This project is comprised of four parts/problems and we have used one uniform framework to solve all the problems.

Framework

Configuration.h

This header file consists of all the virtual functions common to all the project parts.

Functions

- checkGoal- This check whether the goal has been achieved
- enque- This function adds elements to the queue
- deque- This function removes elements to the queue
- isEmpty- This function checks whether the queue is empty
- insert- Adds elements to the map
- nextmove- Provides a set of next moves based on the current state
- firstmove- Checks whether goal has been achieved on the first move
- SolutionMoves- Iterates through the map to get a solution
- display- Prints the solution with the help of solution moves
- writetofile- Writes the output to the file

Puzzle.h

This header file contains the main breadth first search algorithm required to solve all solution and is common to all

Functions

- BFS-The breadth first search algorithm

Part 1- Clock Problem

Header File

- Configuration.h
- Puzzle.h
- Clock.h

Executable File

- Clock.cpp

Clock.h

This file header implements the virtual functions from Configuration.h used to solve the clock problem

Clock.cpp

This is the main executable for the clock problem

Input- It takes command line arguments as three integers

- number of hours on dial
- current clock time
- true time

E.g. 12 3 10

The above represents a 12 hour clock with current time as 3 and the correct time as 10

Part 2- Water Jug Problem

Header File

- Configuration.h
- Puzzle.h
- Water.h

Executable File

- Water.cpp

Water.h

This file header implements the virtual functions from Configuration.h used to solve the water jug problem.

Water.cpp

This is the main executable for the water jug problem

Input- The program takes command line arguments specifying the initial state of the problem.

- The first command line argument is an integer representing the desired amount of water
- The rest of the command line arguments are integers specifying the capacities of the containers.

E.g. 4 3 5

The above represents the goal to be 4 liters of water using two jugs of 3 and 5 liters respectively

Part 3-Lloyds Problem

Header File

- Configuration.h
- Puzzle.h
- Slider.h

Executable File

- Slider.cpp

Slider.h

This file header implements the virtual functions from Configuration.h used to solve the lloyds problem

Slider.cpp

This is the main executable for the llyod problem

Input- The input could either be command line inputs (1) or input be taken from a file (2)

1. Command line Inputs
    - The user can specify the height and width of the puzzle.
    - The user then specifies the input matrix.
2. Input from a file
    - The name of the input file to read for the initial configuration data. If this name is "-" then the initial configuration data is read from the standard input.
    - The name of the output file where the solution is to be written. If this name is "-" then the solution is written to the standard output.