**aCS 438 – Introduction to Artificial Intelligence**

**HW1 – Block World**

**The Background**

Many areas of Computer Science use simple, abstract domains for both analytical and empirical studies. For example, an early AI study of planning and robotics (STRIPS) used a block world in which a robot arm performed tasks involving the manipulation of blocks.

For this assignment, you will write a C++ program (visual studio 2017) that models a simple block world under certain rules and constraints. Rather than determine how to achieve a specified state, you will “program” a robotic arm to respond to a limited set of commands.

**The Problem**

The problem is to parse a series of commands that instruct a robot arm in how to manipulate blocks that lie on a flat table. Initially there are *n* blocks on the table (numbered from 0 to *n*-1) with block *bi* adjacent to block *bi*+1 for all 0 i < n-1 as shown in the diagram below:

|  |
| --- |
|  |

**Figure:** Initial Blocks World

The valid commands for the robot arm that manipulates blocks are described below. Note that A and B are block numbers:

* move A onto B

Puts block A onto block B after returning any blocks that are stacked on top of blocks A and B to their initial positions.

* move A over B

Puts block A onto the top of the stack containing block B, after returning any blocks that are stacked on top of block A to their initial positions.

* pile A onto B

Moves the pile of blocks consisting of block A, and any blocks that are stacked above block A, onto block B. All blocks on top of block B are moved to their initial positions prior to the pile taking place. The blocks stacked above block A retain their order when moved.

* pile A over B

Puts the pile of blocks consisting of block A, and any blocks that are stacked above block A, onto the top of the stack containing block B. The blocks stacked above block A retain their original order when moved.

* quit

Terminates manipulations in the block world.

Any command in which A = B or in which A and B are in the same stack of blocks is an illegal command. All illegal commands should be ignored and should have no affect on the configuration of blocks.

**The Input (from file blockIn.txt)**

The input begins with an integer n on a line by itself representing the number of blocks in the block world. You may assume that 0 < n < 25. The number of blocks is followed by a sequence of block commands, one command per line. Your program should process all commands until the quit command is encountered. You may assume that all commands will be of the form specified above. There will be no syntactically incorrect commands.

**The Output (to file blockOut.txt)**

The output should consist of the final state of the blocks world. Each original block position numbered i (0 i < n, where n is the number of blocks) should appear followed immediately by a colon. If there is at least a block on it, the colon must be followed by one space, followed by a list of blocks that appear stacked in that position with each block number separated from other block numbers by a space. Don't put any trailing spaces on a line.

There should be one line of output for each block position (i.e., n lines of output where n is the integer on the first line of input).

**Sample Input**

10

move 9 onto 1

move 8 over 1

move 7 over 1

move 6 over 1

pile 8 over 6

pile 8 over 5

move 2 over 1

move 4 over 9

quit

**Sample Output**

0: 0

1: 1 9 2 4

2:

3: 3

4:

5: 5 8 7 6

6:

7:

8:

9:

You can also run *BW.exe* to try additional test cases,

**Due Time**

This assignment is worth 100 points and is *due Thursday (1/17/2018) at Class time.* Submit a zip file containing your entire project on Moodle by the due time. A hardcopy of the first page of your program is recommended but not required. Be sure to include necessary comments which includes (but is not limited to) your name, problem description, due date, etc. The final deadline is Friday (1/18/2018) at 5:00 PM after which no program will be accepted.