Function drawUpDown

* Parameters: canvas[], pen\_status, bold\_status, num\_characters, direction, list, command\_number
* Return: nothing
* Logic
  + If pen\_status = false (pen up)
    - for loop <= num\_characters
      * move array pointer up/down one row
  + if pen\_status = true (pen down)
    - for loop <= num\_characters
      * move array pointer up/down one row
      * if bold = true
        + write # to file
      * else
        + peek at character
        + if character != #

write \* to file

Function drawLeftRight

* Parameters: canvas[], pen\_status, bold\_status, num\_characters, direction, list, command\_number
* Return: nothing
* Logic
  + If pen\_status = false (pen up)
    - for loop <= num\_characters
      * + move array pointer left/right on row using pointer to array to traverse the array
  + if pen\_status = true (pen down)
    - if direction = left
      * move array pointer left on row = num\_characters
        + turns left drawing into right drawing
    - for loop <= num characters
      * if bold = true
        + write # to array
      * else
        + point to character at index
        + if character != #

write \* to array

* + - if direction = left
      * move array pointer left on row = num\_characters

Function printPicture

* Parameters: canvas []
* Return: nothing
* Logic:
  + Loop – while < strlen(canvas)
    - Print contents to console
  + Print 2 blank lines (for buffer between prints)

Function printFile

* Parameters: ofstream, canvas
* Return: nothing
* Logic:
  + Loop – while < strlen(canvas)
    - Print contents to output file

Function Main

* Connect istream to commands.txt
* Connect ofstream to paint.txt
* Loop – while !EOF
* Loop – while linked list next pointer is not null
  + command\_number++
  + Read character
  + If char = 1
    - Pen\_status = false (up)
  + Else if char = 2
    - Pen\_status = true (down)
  + Else if char = 3
    - Skip comma
    - Read direction
    - Skip comma
    - Read num
    - If direction = N/S
      * drawUpDown(canvas, pen, bold, num, direction, list, command\_number)
    - else
      * drawLeftRight(pen, bold, num, direction, list, command\_number)
  + else if char = 4
    - printPicture
  + else if char = B
    - bold = true
  + else
    - bold = false
* printPicture
* printFile
* close files

Grid Class

class Grid

{

private:

Gridnode\* head;

public:

Grid();

Grid(Gridnode);

~Grid();

}

Basenode class

class Basenode

{

private:

int row;

int col;

char draw;

public:

Basenode();

Basenode(int, int, char);

~Basenode();

}