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
      * if we are not at first or last row
        + move array pointer up/down one row
      * else
        + delete = 1;
        + deleteCommand(list, delete, command\_number)
  + 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
      * if(direction = left)
        + border = 0 + (current position / 50) \* (51)
      * else
        + border = 49 + (current position / 50) \* (51)
      * if (border - current position) >= num\_characters
        + move array pointer left/right on row using pointer to array to traverse the array
      * else
        + delete = 1;
        + deleteCommand(list, delete, command\_number)
  + 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
  + orderCommands(istream, list)
* 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
    - printFile
  + else if char = B
    - bold = true
  + else
    - bold = false
* printPicture
* printFile
* close files

Function orderCommands

* Paramters: istream, list
* Return: none
* Logic:

A structure will have 4 members that will contain pen status, bold status, direction and number of characters

struct commands

{

int pen\_status;

int bold\_status;

char dir;

int numChars;

commands \*next;

};

For the ordered linked list, make new node and have the pointer node hold the alphabetic identifier in the line of the input file. If there is nothing in the linked list, put that in as the first node. Then for every node afterwards, compare the identifiers and insert to create the order linked list.

Function deleteCommand

* Parameters: list, badCommand, commandNum
* Return: none
* Logic:
  + if badCommand == 1
    - create two nodes to traverse list
    - when first node points to node to be deleted
    - second node then points to first node->next
    - then use delete operation