1. Some obstacles that I overcame are trying to pass the distance from the distance function into other functions. It was challenging since the distance function returns error numbers but not the distance value. Another obstacle was when I fixed one of the errors, it made another error. So, going through the loop of errors after error was mentally hard. Time managing this project during the midterm season was also very hard.
2. Pseudocode

distanceFunction(commandstring, i)

declare char c as the character at i index in the commandString

Declare boolean isNegative and set it to false

Do the following

Declare char c as the character at i index in the commandString

If c isn’t a number or a negative sign or previous character before c isn’t an alphabet or a negative sign

Return error 1 to indicate syntax error

If c is a digit

Set distance to 0

When distance is less than 30 and index is within the commandstring and chracter at the index is a digit

grab the distance

increment the index

if isNegative boolean is true

multiply the distance by -1

return 0 indicating everything went well

if c is a negative sign and distance is 0 and not Negative

set isNegative to true

go onto next character

do the above again if c is negative or a digit

if none of the above was true, return 1 to indicate syntax error

parseCommandFunction

if commandstring at i index is a space and last index isn’t an alphabet and next index isn’t a number

go onto next character

return 1 to indicate something is wrong

if commandstring at i index is a v/V

set direction to vertical

if next index is not negative and not a digit as well

go onto next character

return 1 to indicate something is wrong

go onto next character

call distanceFunction using the commandString and i, get distance back

if plotline function is returning false

decrement the i

while index at commandstring isn’t h/H or v/V

decrement the i until it finds h or v

set badposition to index

return 3 to indicated that this is wrong

return 0 if none of this matches

if index is h/H

set the direction to horizontal

if next index isn’t negative or a digit

go onto next character

return 1 to indicate something is wrong

call distance function, pass on commandstring and index and get distance back

if plotline is false

go back one character

when the character isn’t h/H or v/V

decrement the index until the character is

if index is less than 0

break out of the loop

set bad position to the current index

return 3 to indicate this is wrong

return 0 to indicate it ran correctly

if commandstring at current index is f/F

set mode to foreground

increment the i

set plotChar to next index char

if plotChar isn’t printtable

return 2 to indicate character isn’t printable

if not

increment the i

return 0 to indicate everything’s alright

if commandString at current index is b/B

set mode to background

increment the index to detect the next character

set the plotchar to the index character

if plotchar isn’t printtable

return 2 to indicate character isn’t printable

if not

increment the i

return 0 to indicate everything’s alright

if commandString at current index is c/C

clear the Grid

increment the index

set plotchar to default \*

set mode to foreground

return 0 to indicate everything’s alright

if nothing matches above

return 1 to indicate a syntax error

PerformCommandsFunction

set needed variables such as index, row, column, direction as the default values

do these

if the mode isn’t foreground or background

increment the i and return 2 to indicate something is wrong

if commandstring is empty

break out of the loop, exit the program

call parsecommandstring function

if parsecommandstring is returning 1

set badPos to current index and return 1 to indicate error

if parsecommandstring is returning 3

this function returns 3 too

if parsecommandstring is returning 2

this function returns 2 as well

if direction is horizontal

increase the column by amount of distance as the starting position of next command

if direction is vertical

increase the row by amount of distance as the starting position of next command

while index hasn’t gone through the whole command string

return 0 indicating everything went smoothly

Plotline function

made boolean Valid and set it to false

if either r and c aren’t in grid and fgbg is either foreground or background and plotchar is printable

return false

if direction is horizontal and the whole line is in grid

return true

if direction is vertical and the whole line is in grid

return true

if not

return false

set r to newrow and c to newcolumn

if foreground or space is at the position and passed the cases above

print out the first newrow and new column using plotChar

loop through the distance

if direction is horizontal and distance is positive

newcol will be the one right to last one

if direction is horizontal and distance is positive

newcol will be the one left to last one

if direction is vertical and distance is positive

newrow will be the one below the last one

if direction is vertical and distance is negative

newrow will be the one above the last one

if foreground or space is at the position and passed the cases above

print out the newcol and newrow above, using the plotchar

return true to indicate everything is running well

TEST DATA

| Tests | Reasons |
| --- | --- |
| h29V29 | demonstrate the program reads horizontal and vertical command (uppercase and lowercase) and edge cases well |
| v2b h12fHh1fih0 | demonstrates the program parses through the string and can differentiate between what to do when faced with foreground background commands and horizontal vertical commands |
| h28h-29 | demonstrate the program knows out of grid error for horizontal command |
| v25v-29 | demonstrate the program knows out of grid error for vertical command |
| c | program can clear the grid |
| ch | program can recognize syntax error (no number following horizontal command) |
| b@h5fLh-3 | demonstrates the program can handle background and foreground well |
| v5h-125 | demonstrate program will return error for digits greater than 30 |
| v10h 10 | demonstrate program will not handle if there is invalid syntax |
| h5b@h5ch1b!h-1 | demonstrates that clear command sets mode and character back to default values, but the position stays |
| FH8 | demonstrates that the program returns an error when there is a dangling command |

My program handles all of the above test cases well.