**Project 3: CS31**

**Obstacles I overcame:**

One notable obstacle I overcame while writing my program for this assignment was the fact that I had a fundamental misconception about *break* vs *continue* statements. I believed that break and continue statements were essentially the same and thus you could use them interchangeably. Furthermore, I believed that *break* statements could be used to “break” out of *if* statements even though if statements were not loops. This led to a long and frustrating time where my performCommands function would not parse the string properly, breaking continuously in places I did not actually want. Another notable obstacle I overcame was designing the function that would output the distance after an H or V command. I wrote an entire command up that worked, but when I got to implementing the error commands, I found that it did not work at all. I had to then redesign the entire function from the ground up so that it could handle reporting commands which took a while and was equally frustrating. I learned to deal with frustration quite a lot in this project as little things wouldn’t work or would stop working due to other incorrect changes I made. **Last minute addition:** I was going to turn my project in right now when I found that I had copy pasted some code that were surrounded by one parenthesis on one side and nothing on the other, yielding an error in only a very select few cases. This would have caused the loss of some points so I am glad I found it.

**Design of the Program**:

*User inputs command string*

***performCommands Function:***

*repeatedly*

*if character is letter*

*upper/lower case C:*

*if character after C isn’t a plotting command letter*

*syntax error*

*if no error then reset the grid*

*upper/lower case F:*

*upper/lower case B:*

*set mode to foreground if F*

*set mode to background if B*

*use checkCharacter function to set plotCharacter to character after the letter*

*upper/lower case H:*

*if no value after H*

*syntax error*

*set direction to horizontal*

*use findDistance function to find distance (within 3 characters after the letter)*

*if there was a character where a digit should be*

*syntax error*

*check if the distance would plot inside grid*

*plot the line*

*check if plot the line is true*

*if false, syntax error*

*otherwise, add distance to column and return new column value*

*upper or lower case V:*

*if no value after V*

*syntax error*

*set direction to vertical*

*use findDistance function to find distance (within 3 characters after the letter)*

*if there was a character where a digit should be in the distance*

*syntax error*

*check if the distance would plot inside grid*

*check if plot the line is true*

*if false, syntax error*

*otherwise, add distance to row and return new row value*

*if letter is none of the above:*

*syntax error*

*digits or characters outside plotting command*

*syntax error*

*return true;*

***plotline Function:***

*if the mode, character, or direction of the line are incorrect (according to the command)*

*return error*

*if the distance specified is positive*

*return error if distance is greater than the grid*

*if the direction is horizontal*

*check if distance plots inside grid*

*based on the mode, plot the characters*

*if direction is vertical*

*do the same as above but vertically*

*if the distance specified is negative*

*do the same as above but make lines going to the left or up depending on the mode*

*return true*

**Test Data:**

|  |  |
| --- | --- |
| **F#H+25H?V3!** | If there is a + sign after the H/V command |
| **B@H** | If there are no more characters after H/V commands |
| **C12** | If there are digits after a C command |
| **Q3H5V8** | If there is a letter that is not part of the plotting commands |
| **V03c H123#** | If there is a space after C command where there should be a letter |
| **H18H-123#** | If a number starts a command |
| **H5H-1-2** | If a character starts a command |
| **FH8** | If the same letter as a plotting command is used as a character in a F/B command, digits after are erroneous |
| **H--** | If there are two minus signs in a H/V command, error at second minus sign |
| **H-V17** | If there is a minus after H/V with no digit |
| **H / V / F / B** | If there is a single plotting command and characters are expected afterwards |
| **C** | If C is the only command |
| **H-10** | If H or V tries to plot outside the grid |
| **H5V4H5V4** | If two or more plotting commands in a row |
| **H0** | If H or V have 0 distance |
| **H5H-5** | If you try to reverse distance without changing mode |
| **H5F@H-5** | If you reverse distance after changing the mode |
| **Q** | If command is a random letter alone |
| **3** | If command is a number alone |
| **-** | If command is a character that’s not number or letter alone |
| **h5 / v5 / f@ / b@** | If command has lower case plotting commands |
| **H5V86F** | If there are syntactical issues and there are issues where it tries to print outside the grid, print the syntactical issue even if it is after the other issue |
| **H5v-10** | Valid command followed by command that would go outside the grid |
| **H- / V-** | If minus sign with no digits after, there is an error where it expects a digit to be |
| **H$H30** | If there is a syntax error followed by performance error |
| **H$H#** | If there are multiple syntax errors, consider only the first two |
| **F@4** | If there is a number after F/B command |
| **H25,H-10** | If there is a comma present in between commands |
| **H25 H-10** | If space in between commands |
| **V86H50f@** | If there are two performance command errors, it should return the first performance error |
| **H00** | If there are two 0s |
| **‘ ’ (space character)** | If there is a space alone |
| **f\*v5v-5b h1v1f\*h0b h1v1f\*h0b h1v1f\*h0b h1v1f\*h0b h1v1f\*h0v-5b h2f\*v5b h2v-5f\*v5v-3h5v3v-5b h2f\*v5v-3h5v3v-5h-5b h7f\*v5h5v-5h-5** | Make sure that both upper and lower case commands work and that the program is handling longer command strings just as well |
| **cf h10f\*v2b h1v-2h3f\*v2b h3v2f/h0b v1h-1f/h0b v1h-2f-h-4b v-1h-2f\h0b v-1h-1f\h0** | Making sure it works with nontraditional characters like the / and \ as well |
| **H\t (don’t type in \t, press tab instead)** | Making sure that isprint returns false for this |