ISSUES: 1. The period is supposed to cause a number to print out with a space following it and this does not do that.

2. Popping the empty stack should result in a 0 being pushed onto the stack

3. Missing file name at top upon opening a file

4. Unable to just hit save upon opening a file

5. Displays whole path to a file in title bar

6. Missing program area label

7. When you open a file and alter it, then hit save file, it prompts you just as it does for save as or if the file does not exist when it should just save.

8. not sure if step button should be there

9. issue of adding new line upon saving a file..

May have issue where you typed “abc123!” user may think they need to put “” in

IDENTIFIER:

TEXTBOX-DISPLAY-TEST

DESCRIPTION:

Ensure that upon start up, the system displays a graphical user interface with three textboxes labeled “Program Area”, “Stack”, and “Output”

PRECONDITIONS:

1. Ensure that you have the JBefunge-master file downloaded.
2. Open the JBefunge-Master folder.
3. Open the scr folder
4. Open the run Shell Script file.

EXECUTION STEPS:

1. Once the program starts, visually check to see if there are three text boxes.
2. Visually check to see if the label “Program Area” exists for one of the textboxes.
3. Visually check to see if the label “Stack” exists for one of the textboxes.
4. Visually check to see if the label “Output” exists for one of the textboxes.
5. Visually check to make sure no two names correspond to any textbox.

POSTCONDITIONS:

1. The labels “Program Area”, “Stack”, and “Output” exist for 3 separate textboxes.

Output: The system displays 3 textboxes, with 1 labeled “Stack:”, and another labeled “Output:”, but it is missing the label “Program Area” for the third textbox.

IDENTIFIER:

TEXTBOX-EDITING-START-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable before JBefunge code is run.

PRECONDITIONS:

1. The system is running.
2. All textboxes are empty at the start of the program.

EXECUTION STEPS:

1. Click inside the top textbox and enter “abc123!”
2. Click inside the middle “Stack” textbox and enter “abc123!”
3. Click inside the lower “Output” textbox and enter “abc123!”

POSTCONDITIONS:

The top textbox displays “abc123!”, all others are empty.

Output: should work fine

IDENTIFIER:

TEXTBOX-EDITING-DURING-RUN-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable

while JBefunge code is running using the “Run” option.

PRECONDITIONS:

1. The system is running.
2. All textboxes are empty at the start of the program.
3. Open the file “Fizzbuzz.bf”.

EXECUTION STEPS:

1. Press “Run” to run the program.
2. Left click inside the top textbox and enter “steelers” 1 line under bottom “^”.
3. Visually verify that “steelers” was entered into the program area textbox.
4. Restart the system and do the preconditions again.
5. Press “Run” to run the program.
6. This time, left click anywhere inside the middle “Stack” textbox and enter “steelers”
7. Visually verify that “steelers” was not entered into “Stack” textbox.
8. Restart the system and do the preconditions again.
9. Press “Run” to run the program.
10. Left click anywhere inside the lower “Output” textbox and enter “steelers”
11. Visually verify that “steelers” was not entered the “Output” textbox.

POSTCONDITIONS:

1. Only the Program Area textbox should have the code plus “steelers” in it. The “Stack” and “Output” textboxes should have displayed what “Fizzbuzz.bf” displays without a user trying to alter the textboxes.

Output: should work fine

IDENTIFIER:

TEXTBOX-EDITING-DURING-MOSEY-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable

while JBefunge code is running using the “Mosey” option.

PRECONDITIONS:

1. The system is running.
2. All textboxes are empty at the start of the program.
3. Open the file “Fizzbuzz.bf”.
4. Press “Run” to run the program.

EXECUTION STEPS:

1. Press “Mosey” to run the program.
2. Left click inside the top textbox and enter “steelers” 1 line under bottom “^”.
3. Visually verify that “steelers” was entered into the program area textbox.
4. Left click anywhere inside the middle “Stack” textbox and enter “steelers”
5. Visually verify that “steelers” was not entered into the “Stack” textbox.
6. Left click anywhere inside the lower “Output” textbox and enter “steelers”
7. Visually verify that “steelers” was not entered into the “Output” textbox.

POSTCONDITIONS:

1. Only the Program Area textbox should have the code plus “steelers” in it. The “Stack” and “Output” textboxes should have displayed what “Fizzbuzz.bf” displays without a user trying to alter the textboxes.

IDENTIFIER:

TEXTBOX-EDITING-DURING-WALK-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable

while JBefunge code is running using the “WALK” option.

PRECONDITIONS:

1. The system is running.
2. All textboxes are empty at the start of the program.
3. Open the file “Fizzbuzz.bf”.

EXECUTION STEPS:

1. Press “Walk” to run the program.
2. Left click inside the top textbox and enter “steelers” 1 line under bottom “^”.
3. Visually verify that “steelers” was entered into the program area textbox.
4. Left click anywhere inside the middle “Stack” textbox and enter “steelers”
5. Visually verify that “steelers” was not entered into the “Stack” textbox.
6. Left click anywhere inside the lower “Output” textbox and enter “steelers”
7. Visually verify that “steelers” was not entered into the “Output” textbox.

POSTCONDITIONS:

1. Only the Program Area textbox should have the code plus “steelers” in it. The “Stack” and “Output” textboxes should have displayed what “Fizzbuzz.bf” displays without a user trying to alter the textboxes.

IDENTIFIER:

TEXTBOX-EDITING-BETWEEN-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable after the stop is hit when JBefunge code has been running.

PRECONDITIONS:

1. The system is running.

EXECUTION STEPS:

POSTCONDITIONS:

N/A

IDENTIFIER:

TEXTBOX-EDITING-POST-TEST

DESCRIPTION:

Ensure that only the Program Area textbox is user-editable after a JBefunge program has ran.

PRECONDITIONS:

1. The system is running.
2. All textboxes are empty at the start of the program.
3. Go to file in the menu bar, then choose open, and open the file HelloWorld.bf.
4. Press run at the bottom of the program on the far left of all the buttons.

EXECUTION STEPS:

1. Click inside the top textbox and enter “abc123!”
2. Click inside the middle “Stack” textbox and enter “abc123!”
3. Click inside the lower “Output” textbox and enter “abc123!”

POSTCONDITIONS:

The top textbox displays “64+"!dlroW ,olleH">:#,\_@abc123!”. The “Stack” textbox displays “[]”. The “Output” textbox displays “Hello, World!”.

Output: should work fine

IDENTIFIER:

FILE-MENU-TEST

DESCRIPTION:

Ensure that there is the menu group “File” with the menu items “Open File”, “Save File”, “Save As”, and “Quit”.

PRECONDITIONS:

1. The system is running.

EXECUTION STEPS:

1. First check the menu bar to see if the “File” menu group exists.
2. Click on the “File” menu group.
3. Check to see if “Open File” exists in the menu items.
4. Check to see if “Save File” exists in the menu items.
5. Check to see if “Save As” exits in the menu items.
6. Check to see if “Quit” exits in the menu items.
7. Make sure there are only the above 4 menu items inside the “File” menu group.

POSTCONDITIONS:

1. “Open File”, “Save File”, “Save As”, and “Quit” should be visible within the “File” display group.

IDENTIFIER:

COLOR-MENU-TEST

DESCRIPTION:

Ensure that there is the menu group “Color” with the menu items “Red”, “Yellow”, “Blue”, “Pink”, “Green”, and “Orange”.

PRECONDITIONS:

1. The system is running.

EXECUTION STEPS:

1. First check the menu bar to see if the “Color” menu group exists.
2. Click on the “Color” menu group.
3. Check to see if “Red” exists in the menu items.
4. Check to see if “Yellow” exists in the menu items.
5. Check to see if “Blue” exits in the menu items.
6. Check to see if “Pink” exits in the menu items.
7. Check to see if “Green” exits in the menu items.
8. Check to see if “Orange” exits in the menu items.
9. Make sure there are only the above 6 menu items inside the “Color” menu group.

POSTCONDITIONS:

1. “Red”, “Yellow”, “Blue”, “Pink”, “Green”, and “Orange should be visible within the “Color” menu group.

IDENTIFIER:

OPTIONS-MENU-TEST

DESCRIPTION:

Ensure that there is the menu group “Options” with the checkable menu items “Time Programs” and “Check for End Opcode”.

PRECONDITIONS:

1. The system is running.

EXECUTION STEPS:

1. First check the menu bar to see if the “Options” menu group exists.
2. Click on the “Options” menu group.
3. Check to see if “Time Programs” exists in the menu items and has a check box.
4. Check to see if “Check for End Opcode” exists in the menu items and has a check box.
5. Make sure there are only the above 2 menu items inside the “Options” menu group.

POSTCONDITIONS:

1. “Time Programs” and “Check for End Opcode” should have checkable labels within the “Options” menu group.

IDENTIFIER:

MENU-TEST

DESCRIPTION:

Ensure that there are only three menu groups labeled “File”, “Color”, and “Options”.

PRECONDITIONS:

1. The system is running.

EXECUTION STEPS:

1. First check the menu bar to see if the “File” menu group exists.
2. Then check the menu bar to see if the “Color” menu group exists.
3. Lastly, check the menu bar to see if the “Options” menu group exists.
4. Make sure there are only the above 3 menu groups in the menu bar.

POSTCONDITIONS:

There should be the menu groups “File”, “Color”, and “Options” visible within the application.

IDENTIFIER:

FIRST-SAVE-AS-TEST

DESCRIPTION:

Ensure that the “Save As” function properly saves a file when saving a file for the first time.

PRECONDITIONS:

1. The system is running, and all textboxes are empty.
2. Ensure that a file named “abc.bf” does not already exist in the “JBefunge-master” folder, otherwise delete it.
3. Place “abc123!” in the program area textbox.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Save as” in the menu bar.
3. Once the “Save” pop-up window appears, go to the “JBefunge-master” folder
4. Enter “abc.bf” as the file name in the “File Name” textbox of the “Save” window.
5. Click the “Save” button at the bottom of the “Save” window.
6. Exit the program entirely.
7. Reopen the program.
8. Click on “File” in the menu bar.
9. Click on “Open file”.
10. In the open pop-up window, find the “JBefunge-master” folder and open it.
11. Once inside the folder, find the file “abc.bf”.
12. Click on “abc.bf” and press open on the bottom right of the “Open” window.
13. Upon opening, check to see if “abc123!” is in the program area textbox and all other textboxes are empty.

POSTCONDITIONS:

1. “abc123!” should be displayed in the program area textbox and all other textboxes should be empty.
2. The file “abc.bf” should exist within the “JBefunge-master” folder.

IDENTIFIER:

FIRST-SAVE-TEST

DESCRIPTION:

Ensure that the “Save File” function properly works when saving a file for the first time.

PRECONDITIONS:

1. The system is running and no file has been saved during the current session.
2. Ensure that a file named “abc.bf” does not already exist in the “JBefunge-master” folder, otherwise delete it.
3. Place “abc123!” in the program area textbox.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Save File” in the menu bar.
3. Once the “Save” pop-up window appears, go to the “JBefunge-master” folder.
4. Enter “abc.bf” as the file name in the “File Name” textbox of the “Save” window.
5. Click the “Save” button at the bottom of the “Save” window.
6. Exit the program entirely.
7. Reopen the program.
8. Click on “File” in the menu bar.
9. Click on “Open file”.
10. In the open pop-up window, find the “JBefunge-master” folder and open it.
11. Once inside the folder, find the file “abc.bf”.
12. Click on “abc.bf” and press open on the bottom right of the “Open” window.
13. Upon opening, check to see if “abc123!” is in the program area textbox and all other textboxes are empty.

POSTCONDITIONS:

1. “abc123!” should be displayed in the program area textbox and all other textboxes should be empty.
2. The file “abc.bf” should exist inside the “JBefunge-master” folder.

IDENTIFIER:

OPEN-SAVE-TEST

DESCRIPTION:

Ensure that the “Save File” function properly works when saving a file that you just opened.

PRECONDITIONS:

1. The system is running.
2. A file named “abc.bf” should exist in the “JBefunge-master” folder that has “abc123!” inside the program area textbox when opened.
3. If such file does not exist, create one as specified above and restart the program.
4. Open the file “abc.bf”

EXECUTION STEPS:

1. Click inside the program area textbox and add “steelers” onto the end of “abc123!”.
2. Click on “File” in the menu bar.
3. Click on “Save File” in the menu bar.
4. Exit the program entirely.
5. Reopen the program.
6. Click on “File” in the menu bar.
7. Click on “Open file”.
8. In the open pop-up window, find the “JBefunge-master” folder and open it.
9. Once inside the folder, find the file “abc.bf”.
10. Click on “abc.bf” and press open on the bottom right of the “Open” window.
11. Upon opening, check to see if “abc123!steelers” is in the program area textbox and all other textboxes are empty.

POSTCONDITIONS:

1. “abc123!steelers” should be displayed in the program area textbox and all other textboxes should be empty.
2. The file “abc.bf” should exist within the “JBefunge-master” folder.

Issues: When you open a file and alter it, then hit save file, it prompts you just as it does for save as or if the file does not exist when it should just save.

Once you save the file under some name, if enter some text at the end of a line that already existed, after closing and reopening the program, it places the text on the next line which could be a major issue.

If you save a new file, and then enter new code at the end of the line, this is not an issue though.

Not pushing down one line if on a new line

IDENTIFIER:

SAVE-POST-TEST

DESCRIPTION:

Ensure that the “Save File” function properly works when saving a file that has recently been saved.

PRECONDITIONS:

1. The system is running.
2. A file named “abc.bf” should exist that has “abc123!” inside the program area textbox when opened.
3. Open the file “abc.bf”
4. Save the file “abc.bf” once

EXECUTION STEPS:

1. Click inside the program area textbox and add “steelers” onto the end of “abc123!”.
2. Click on “File” in the menu bar.
3. Click on “Save File” in the menu bar.
4. Exit the program entirely.
5. Reopen the program.
6. Click on “File” in the menu bar.
7. Click on “Open file”.
8. In the open pop-up window, find the “JBefunge-master” folder and open it.
9. Once inside the folder, find the file “abc.bf”.
10. Click on “abc.bf” and press open on the bottom right of the “Open” window.
11. Upon opening, check to see if “abc123!steelers” is in the program area textbox and all other textboxes are empty.

POSTCONDITIONS:

1. “abc123!steelers” should be displayed in the program area textbox and all other textboxes should be empty.
2. The file “abc.bf” should exist within the “JBefunge-master” folder.

IDENTIFIER:

FIRST-OPEN-TEST

DESCRIPTION:

Ensure that the “Open File” function properly works when opening a file directly after starting the program.

PRECONDITIONS:

1. The system is running and no programs have been opened or saved since opening the system.
2. A file named “HelloWorld.bf” should exist inside the “JBefunge-master” folder that has “64+"!dlroW ,olleH">:#,\_@” inside the file.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Open File” in the menu bar.
3. Once the “Open” window appears, open the “JBefunge-master” folder.
4. Once inside the folder, open the “HelloWorld.bf” file by clicking on it and then hitting the open button on the bottom right of the “Open” window.
5. Visually check to see if “64+"!dlroW ,olleH">:#,\_@” appears inside the program area textbox and that all other text boxes are empty.

POSTCONDITIONS:

1. “64+"!dlroW ,olleH">:#,\_@” should be displayed in the program area textbox and all other textboxes should be empty.

IDENTIFIER:

SECOND-OPEN-TEST

DESCRIPTION:

Ensure that the “Open File” function properly works when opening a file after another file has already been opened.

PRECONDITIONS:

1. The system is running.
2. A file named “HelloWorld.bf” should exist inside the “JBefunge-master” folder that has “64+"!dlroW ,olleH">:#,\_@” inside the file.
3. A file named “FizzBuzz.bf” should exist inside the “JBefunge-master” folder.
4. Open the file “FizzBuzz.bf”.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Open File” in the menu bar.
3. Once the “Open” window appears, open the “JBefunge-master” folder.
4. Once inside the folder, open the “HelloWorld.bf” file by clicking on it and then hitting the open button on the bottom right of the “Open” window.
5. Visually check to see if “64+"!dlroW ,olleH">:#,\_@” appears inside the program area textbox and that all other text boxes are empty.

POSTCONDITIONS:

1. “64+"!dlroW ,olleH">:#,\_@” should be displayed in the program area textbox and all other textboxes should be empty.

IDENTIFIER:

RUN-OPEN-TEST

DESCRIPTION:

Ensure that the “Open File” function properly works when opening a file after another file has already been opened and has been ran.

PRECONDITIONS:

1. The system is running.
2. A file named “HelloWorld.bf” should exist inside the “JBefunge-master” folder that has “64+"!dlroW ,olleH">:#,\_@” inside the file.
3. A file named “FizzBuzz.bf” should exist inside the “JBefunge-master” folder.
4. Open the file “FizzBuzz.bf”.
5. Run the file to completion.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Open File” in the menu bar.
3. Once the “Open” window appears, open the “JBefunge-master” folder.
4. Once inside the folder, open the “HelloWorld.bf” file by clicking on it and then hitting the open button on the bottom right of the “Open” window.
5. Visually check to see if “64+"!dlroW ,olleH">:#,\_@” appears inside the program area textbox and that all other text boxes are empty.

POSTCONDITIONS:

1. “64+"!dlroW ,olleH">:#,\_@” should be displayed in the program area textbox and all other textboxes should be unaltered.

IDENTIFIER:

OPEN-LABEL-TEST

DESCRIPTION:

Ensure that the name of the opened file appears at the top of the application after the file is opened.

PRECONDITIONS:

1. The system is running.
2. A file named “HelloWorld.bf” should exist inside the “JBefunge-master” folder that has “64+"!dlroW ,olleH">:#,\_@” inside the file.
3. The title bar of the window should say “UNTITLED” at the start of the application.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Open File” in the menu bar.
3. Once the “Open” window appears, open the “JBefunge-master” folder.
4. Once inside the folder, open the “HelloWorld.bf” file by clicking on it and then hitting the open button on the bottom right of the “Open” window.
5. Visually check to see if “HelloWorld” appears in the menu bar at the top of the window.

POSTCONDITIONS:

1. The title bar of the window should display “HelloWorld”.

DOES NOT SAY THIS UPON OPENING A FILE

IDENTIFIER:

PRESAVE-LABEL-TEST

DESCRIPTION:

Ensure that “UNTITLED” appears at the top of the application when the current program has never been named or opened.

PRECONDITIONS:

1. The system is running, and no programs have been opened or saved during the current session.

EXECUTION STEPS:

1. Visually check to see if “UNTITLED” appears at the top of the window.
2. Enter “abc123” into the program area textbox.
3. Visually check to see if “UNTITLED” still appears at the top of the window.

POSTCONDITIONS:

1. The title bar of the window should say “UNTITLED”.

IDENTIFIER:

POST-SAVE-LABEL-TEST

DESCRIPTION:

Ensure that the name of the just saved file appears at the top of the application.

PRECONDITIONS:

1. The system is running, and no programs have been opened or saved in the current session.
2. The title bar of the window should say “UNTITLED”.

EXECUTION STEPS:

1. Click on “File” in the menu bar.
2. Click on “Save File” in the menu bar.
3. Once the “Save” pop-up window appears, go to the “JBefunge-master” folder.
4. Enter “abc.bf” as the file name in the “File Name” textbox of the “Save” window.
5. Click the “Save” button at the bottom of the “Save” window.
6. Visually check to see if “abc” appears in the menu bar of the window.

POSTCONDITIONS:

1. The title bar of the window should say “abc”.

THIS KIND OF WORKS BUT DISPLAYS WHOLE PATH TO FILE, NOT JUST THE FILE NAME

IDENTIFIER:

RUN-SPEED-TEST

DESCRIPTION:

Ensure that the system has the run speed “Run” which has no pauses during the execution after each opcode.

PRECONDITIONS:

1. The system is running and all textboxes are empty.
2. Open the program “Helloworld.bf” found in the JBefungeMaster folder.

EXECUTION STEPS:

1. Click on “Run” in the bottom of the application window.
2. Visually check to ensure the program is not pausing between opcodes.

POSTCONDITIONS:

1. The program should display “Hello World!” in the “Output” textbox.

IDENTIFIER:

BASE-WALK-SPEED-TEST

DESCRIPTION:

Ensure that the system has the run speed “Walk” which has a 50 millisecond pause with a single opcode.

PRECONDITIONS:

1. The system is running and all textboxes are empty.
2. Type “@” into the program area textbox.
3. Turn on the “Time Program” option found in the “Options” drop down from the menu bar.

EXECUTION STEPS:

1. Click on “Walk” in the bottom of the application window.
2. Visually check to ensure the program is pauses before the single opcode.

POSTCONDITIONS:

1. The “Output” textbox should be empty upon completion.
2. A message box should pop up displaying the amount of time it took to execute the single opcode in microseconds.

IDENTIFIER:

WALK-SPEED-TEST

DESCRIPTION:

Ensure that the system has the run speed “Walk” which has a 50 millisecond pause between each opcode.

PRECONDITIONS:

1. The system is running and all textboxes are empty.
2. Open the program “Helloworld.bf” found in the JBefungeMaster folder.
3. Turn on the “Time Program” option found in the “Options” drop down from the menu bar.
4. Need to first figure out how many opcodes there are for “Helloworld.bf” so use the “Step” button to run through the program to completion and count how many times opcodes there are (107 in total).

EXECUTION STEPS:

1. Click on “Walk” in the bottom of the application window.
2. Visually check to ensure the program is pausing between opcodes.
3. Upon completion, divide the time to execute by the number of opcodes(107).

POSTCONDITIONS:

1. The program should display “Hello World!” in the “Output” textbox.
2. A message box should pop up displaying the amount of time it took to execute the program in microseconds.
3. Divide this output by 107 and move the decimal over 3 places to get the time per opcode on average in milliseconds.

IDENTIFIER:

BASE-MOSEY-SPEED-TEST

DESCRIPTION:

Ensure that the system has the run speed “Mosey” which has a 500 millisecond with a single opcode.

PRECONDITIONS:

1. The system is running and all textboxes are empty.
2. Type “@” into the program area textbox.
3. Turn on the “Time Program” option found in the “Options” drop down from the menu bar.

EXECUTION STEPS:

1. Click on “Mosey” in the bottom of the application window.
2. Visually check to ensure the program is pausing between the single opcode.

POSTCONDITIONS:

1. The “Output” textbox should be empty upon completion.
2. A message box should pop up displaying the amount of time it took to execute the single opcode in microseconds.

IDENTIFIER:

MOSEY-SPEED-TEST

DESCRIPTION:

Ensure that the system has the run speed “Mosey” which has a 500 millisecond pause between each opcode.

PRECONDITIONS:

1. The system is running.
2. Open the program “Helloworld.bf” found in the JBefungeMaster folder.
3. Turn on the “Time Program” option found in the “Options” drop down from the menu bar.
4. Need to first figure out how many opcodes there are for “Helloworld.bf” so use the “Step” button to run through the program to completion and count how many times opcodes there are (107 in total).

EXECUTION STEPS:

1. Click on “Mosey” in the bottom of the application window.
2. Visually check to ensure the program is pausing between opcodes.
3. Upon completion, divide the time to execute by the number of opcodes(107).

POSTCONDITIONS:

1. The program should display “Hello World!” in the output textbox.
2. A message box should pop up displaying the amount of time it took to execute the program in microseconds.
3. Divide this output by 107 and move the decimal over 3 places to get the time waited per opcode on average in milliseconds.

Can test the editing before, while running a program, and after running a program

Could also test during mosey, stop, etc.