REPORT

Reflection:

1. How was this application designed with usability in mind?

    This was designed with usability in mind because it runs off input from the user. The application can be customized based off the user's inputs. No matter what the user inputs (as long as it is a valid response), the application should be able to run within the confines of the program (for example: patterns not exceeding borders, patterns filling as much space as possible, patterns appearing in the correct quadrant, etc,).

2. What was the easiest and hardest part of this assignment?

    The easiest part of this assignment was creating a frame function. This function utilized an even simpler square function, and so when you called it the only parameter needed was the one that determined the size of the individual squares, which then built the grid. The hardest part of the assignment was thinking through the wall_paper() function, testing it, and then troubleshooting it. The math behind the function was a little tricky to visualize, and I would sometimes confuse myself talking through my own program.

3. What did you learn?

    I became much more comfortable with while loops. I also had an opportunity to practice using the print() function to test why something wasn't working. This was particularly handy when testing the wall_paper() function. Finally, I learned why it's important to have a plan and to map out your program before you get started. It's very easy to just start based off your first idea, then continuously add to it as you go. However, when you write down your ideas, do a little math, and talk through your function you have a better understanding of how it works, and what to change when you notice something isn't working.

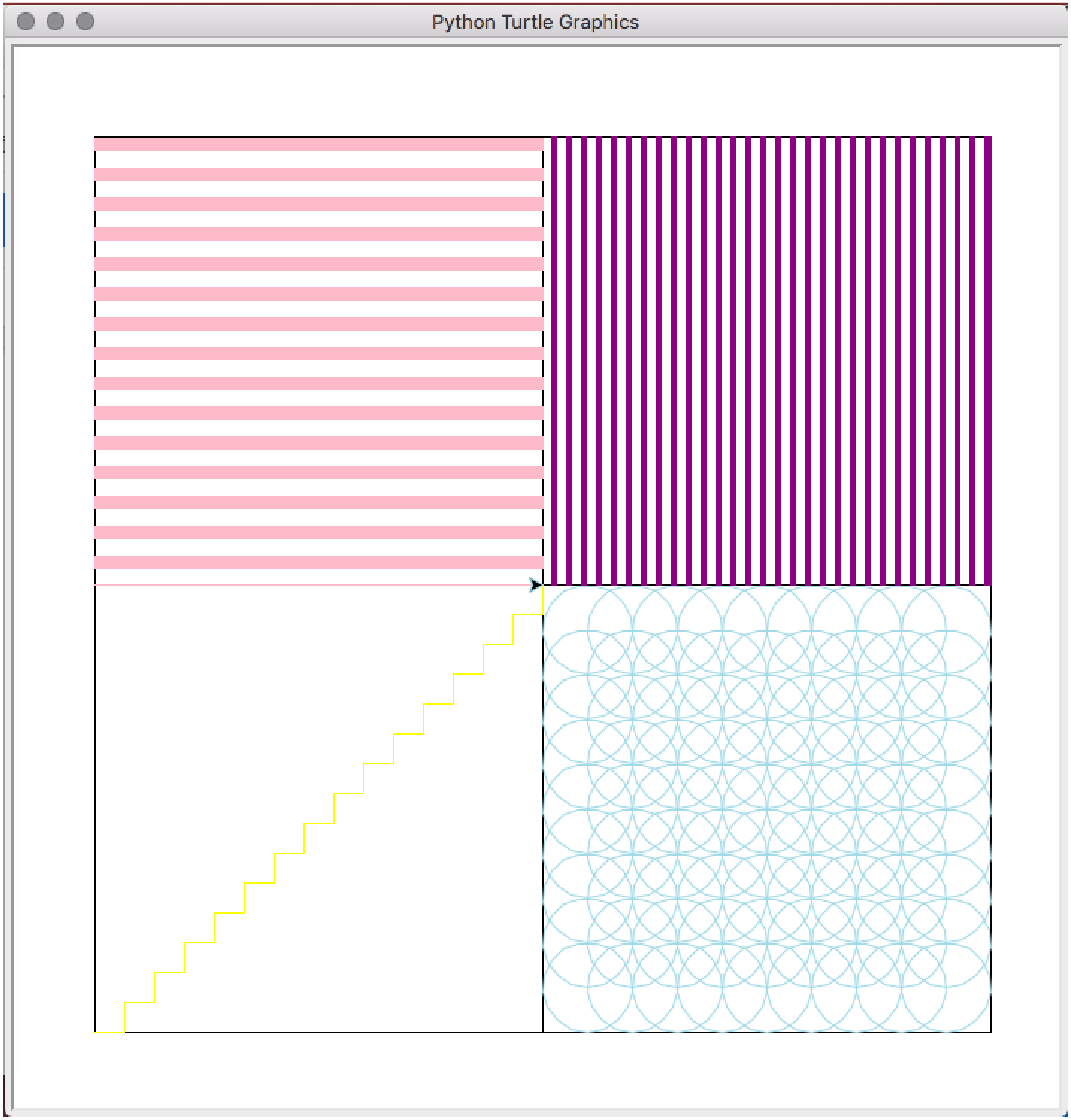
4. What grade would you give yourself?

    I would give myself a 27. The basic functionality of this lab's program took me a really long time to get through, and as a result I didn't have much time to look into extensions (other than colors and validation, which I used in previous labs). However, everything works as intended to the best of my knowledge and I always make sure to keep my coding as clean and organized as possible.

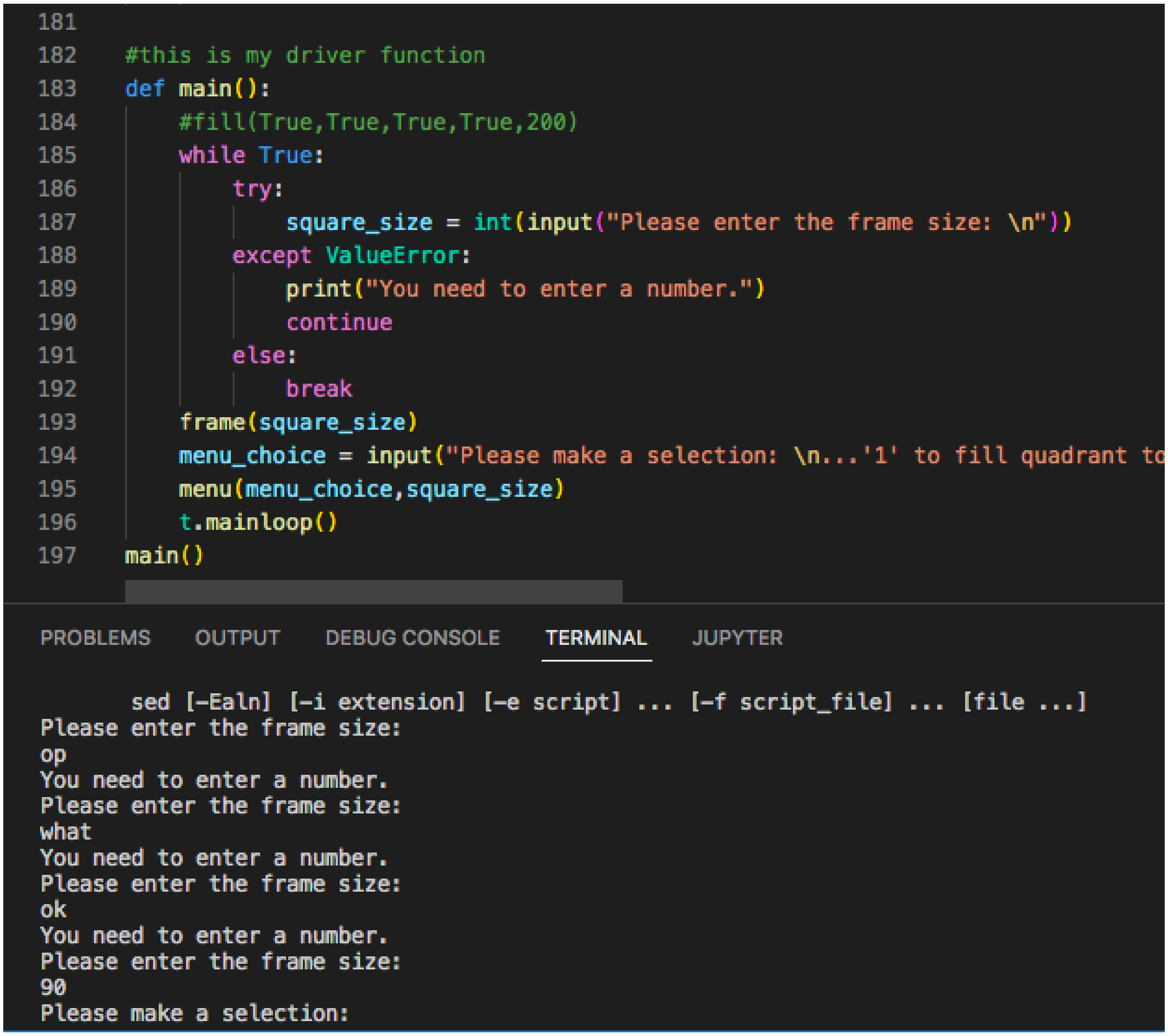
Extension:

1. What extension did you add to the assignment?

    a. My first extension was the basic turtle.pencolor function. I used this to add color to the patterns I made and to make the patterns a little more exciting.



b. My second extension is validation, which is use in two different ways within my program. The first validation happens when the user is asked for a frame size. If the user enters anything that is not a number, the program will ask for a frame input again. The program will ask for a frame input until it receives the correct value type. I did this by using a while loop.

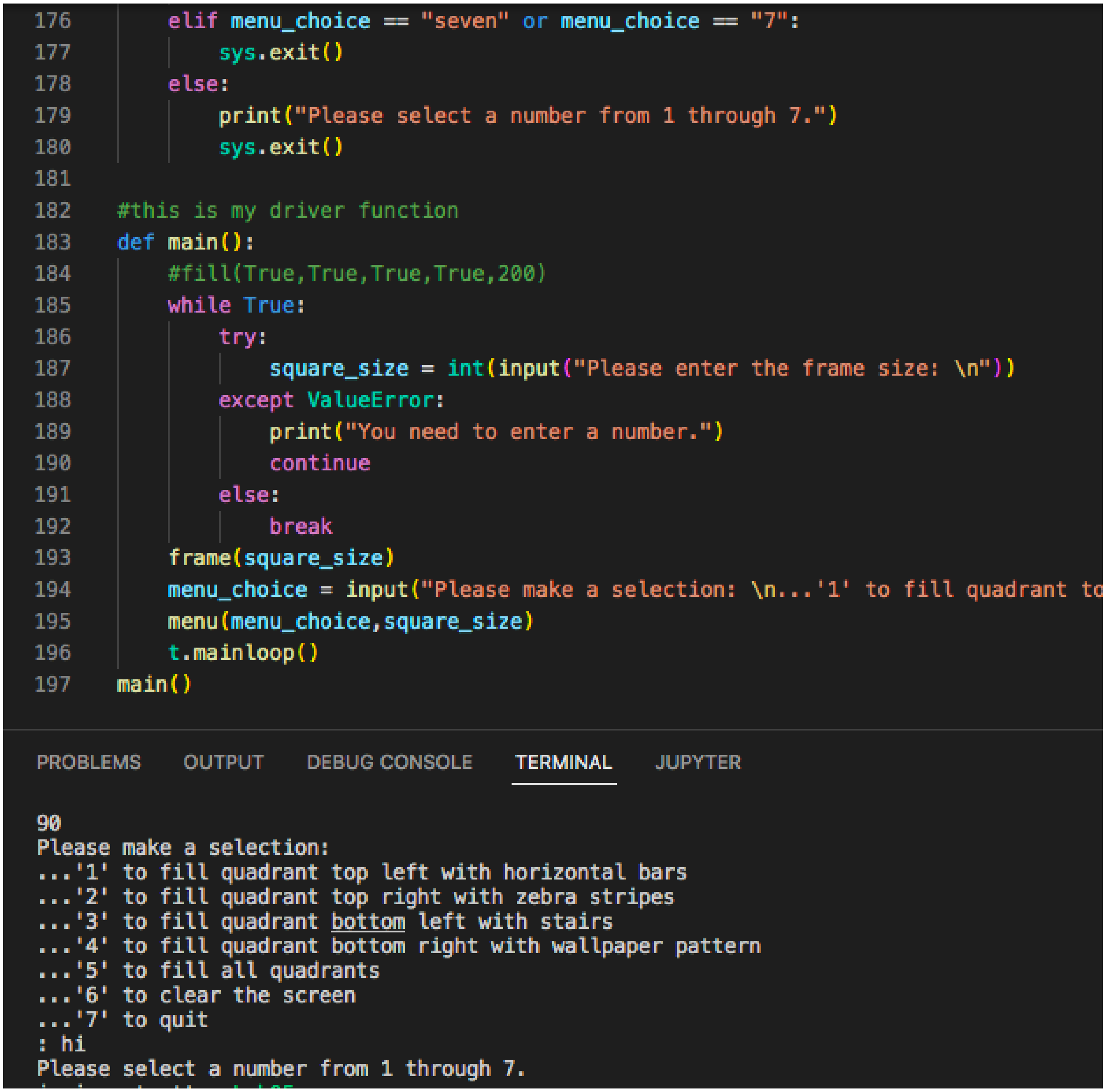


```python

#this is my driver function
def main():
    #fill(True,True,True,True,200)
    while True:
        try:
            square_size = int(input("Please enter the frame size: \n"))
        except ValueError:
            print("You need to enter a number.")
            continue
        else:
            break
    frame(square_size)
    menu_choice = input("Please make a selection: \n...'1' to fill quadrant to
    menu(menu_choice,square_size)
    t.mainloop()
main()
```

```
       sed [-Ealn] [-i extension] [-e script] ... [-f script_file] ... [file ...]
Please enter the frame size:
op
You need to enter a number.
Please enter the frame size:
what
You need to enter a number.
Please enter the frame size:
ok
You need to enter a number.
Please enter the frame size:
90
Please make a selection:
```

The second type of validation is much simpler, and it happens at the second step in the program. The program will ask the user to select from a menu of options what they would like to do (numbered 1-7). The program will accept both the written number (example: three) and the integer (example: 3). If the program receives any input that is not one of the numbers 1-7 it will remind the user to only enter numbers 1-7 and exit the program using sys.exit().



```
    elif menu_choice == "seven" or menu_choice == "7":
        sys.exit()
    else:
        print("Please select a number from 1 through 7.")
        sys.exit()

#this is my driver function
def main():
    #fill(True,True,True,True,200)
    while True:
        try:
            square_size = int(input("Please enter the frame size: \n"))
        except ValueError:
            print("You need to enter a number.")
            continue
        else:
            break
    frame(square_size)
    menu_choice = input("Please make a selection: \n...'1' to fill quadrant to
    menu(menu_choice,square_size)
    t.mainloop()
main()
```

```
90
Please make a selection:
...'1' to fill quadrant top left with horizontal bars
...'2' to fill quadrant top right with zebra stripes
...'3' to fill quadrant bottom left with stairs
...'4' to fill quadrant bottom right with wallpaper pattern
...'5' to fill all quadrants
...'6' to clear the screen
...'7' to quit
: hi
Please select a number from 1 through 7.
```