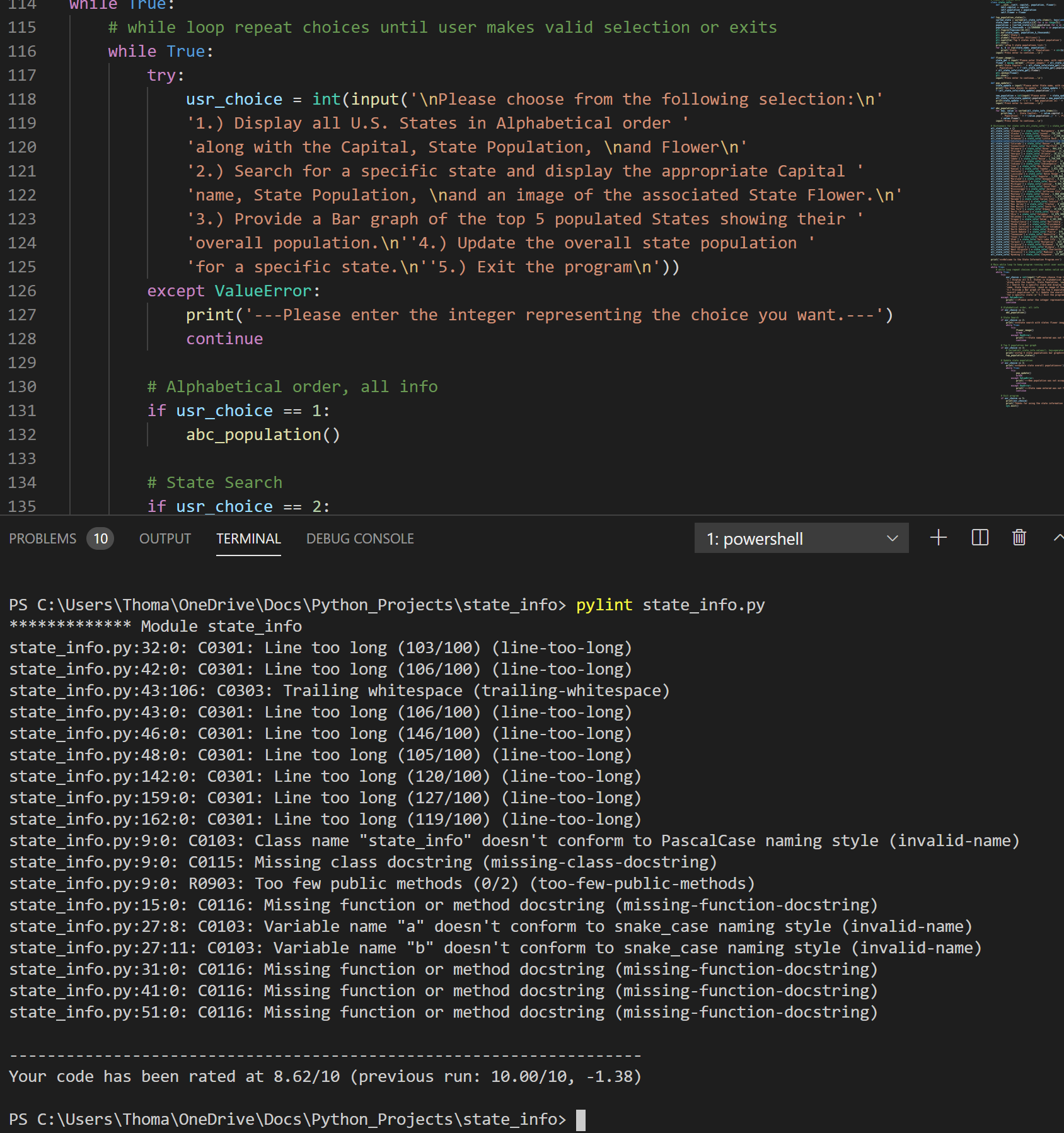
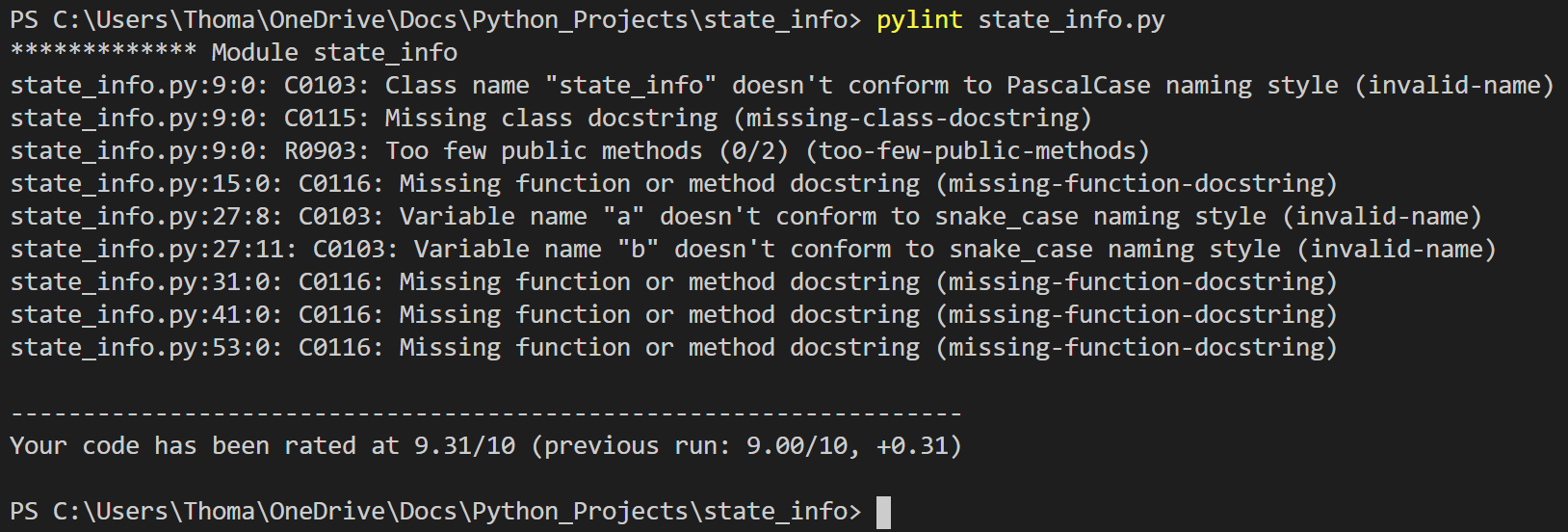
## Pylint Results:

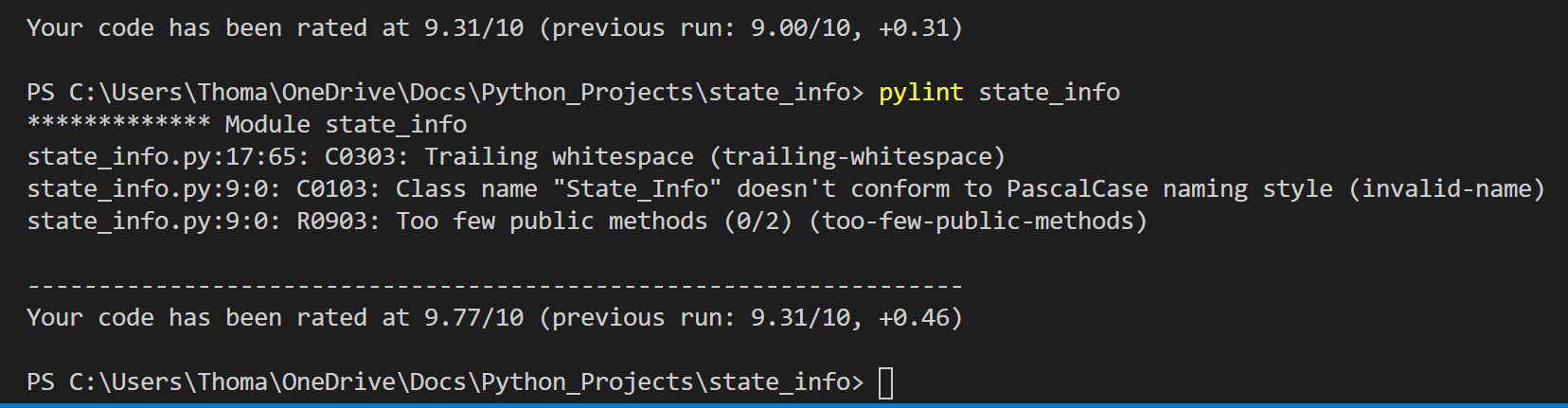
For this lab I was a bit sloppy with my styling and the Pylint results reflect that:



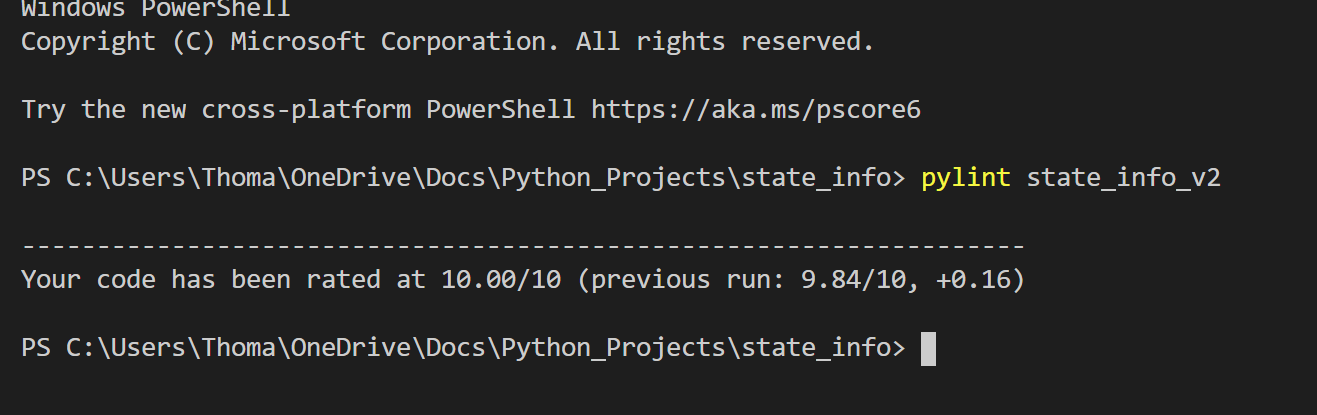
I got rid of all the leading or trailing whitespaces and fixed all the lines that were to long:



Now I need to add docstrings, fix a class name and fix two variable names:



Did not realize PascaCase does not use underscores so had to modify that again, need to fix too few public methods.



I got rid of the too few public methods, by deleting the class StateInfo, that made all of the dictionary’s values objects. I thought this would be best as I could get state info from keys by just using something like .population. However, it turned out to be straight forward to just have the dictionaries keys values be in a list then indexed into for information. Had to update all the dictionary values too.

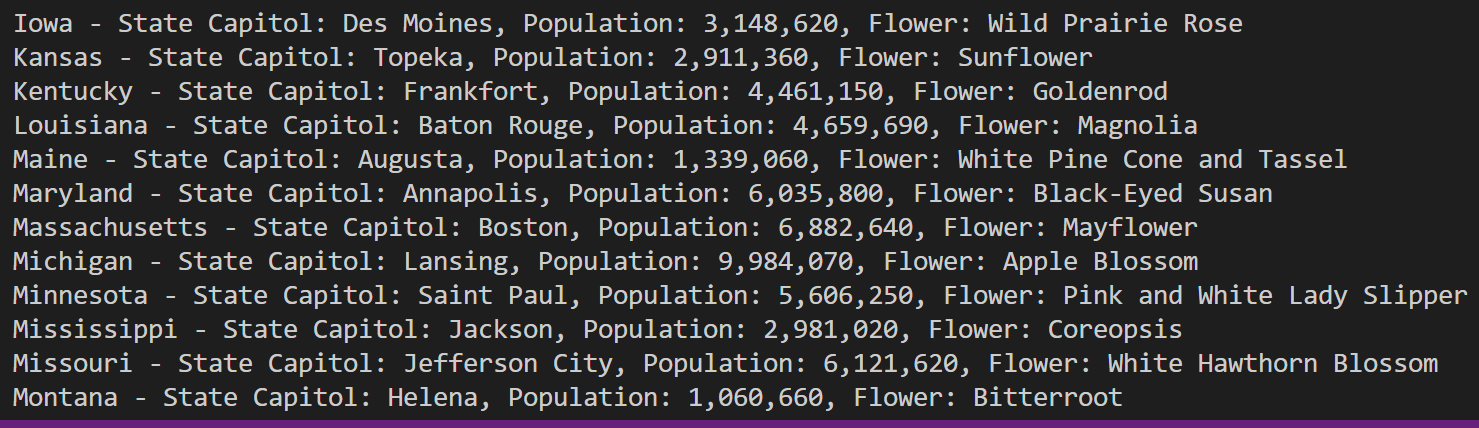
Testing:

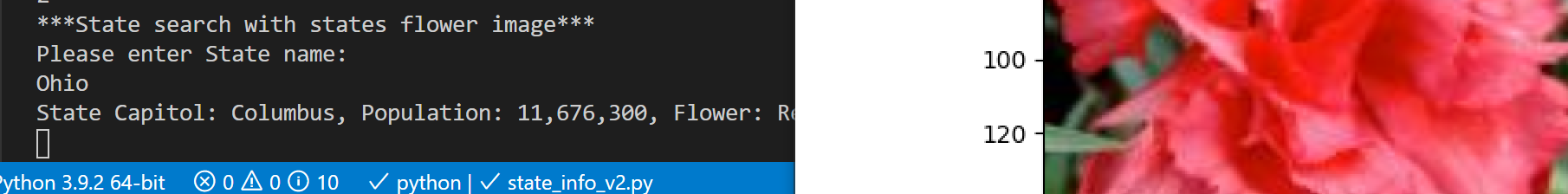
General functionality test:

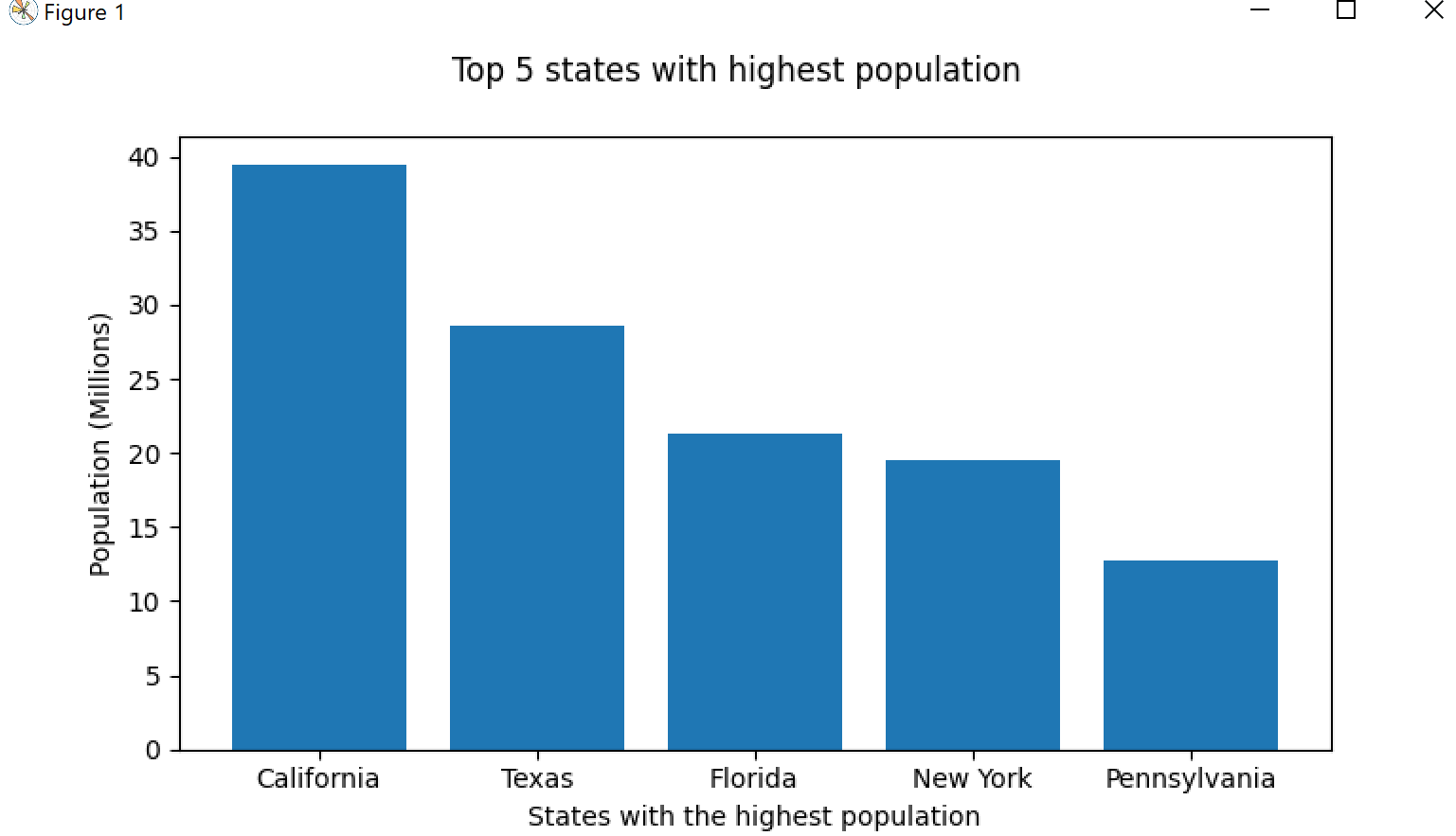
Make sure user can get to all parts of the program with correct input

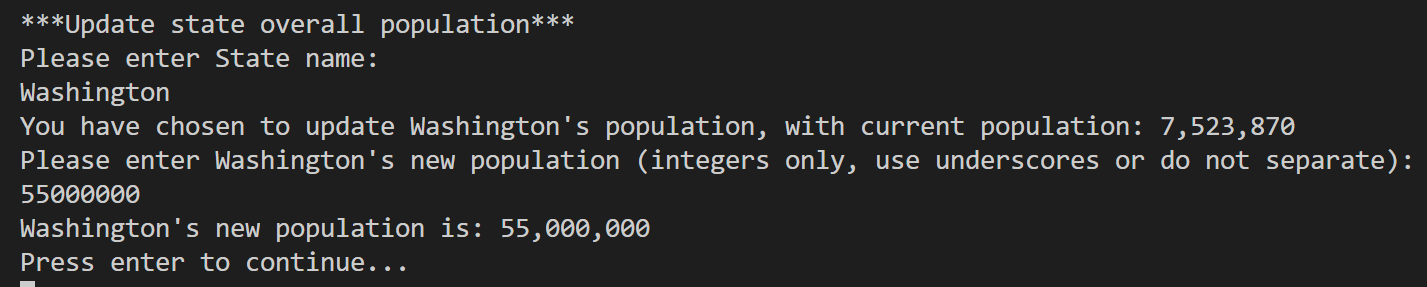
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test # | Input | Expected Output | Output | Pass / Fail |
| 1 | 1, 2, Ohio, 3, 4, Washington, 55000000, 3, 5 | All states in alphabetical order, Ohio’s Information and picture of flower, bar graph, Washington population update, Washington at top of bar graph after update, exit | All states in alphabetical order, Ohio’s Information and picture of flower, bar graph, Washington population update, Washington at top of bar graph after update, exit | Pass |
| 2 | 1, 2, Florida, 3, 4, Michigan, 12000000, 2,Michigan, 5 | All states in alphabetical order, Florida’s Information and picture of flower, bar graph, Michigan population update, Michigan’s new population shown at search, exit | All states in alphabetical order, Florida’s Information and picture of flower, bar graph, Michigan population update, Michigan’s new population shown at search, exit | Pass |
| 3 | 1, 2, Nevada, 3, 4, New York, 80000000, 2, New York, 3, 5 | All states in alphabetical order, Nevada’s Information and picture of flower, bar graph, New York population update, New York new population show, New York at top of bar graph after update, exit | All states in alphabetical order, Nevada’s Information and picture of flower, bar graph, New York population update, New York new population show, New York at top of bar graph after update, exit | Pass |

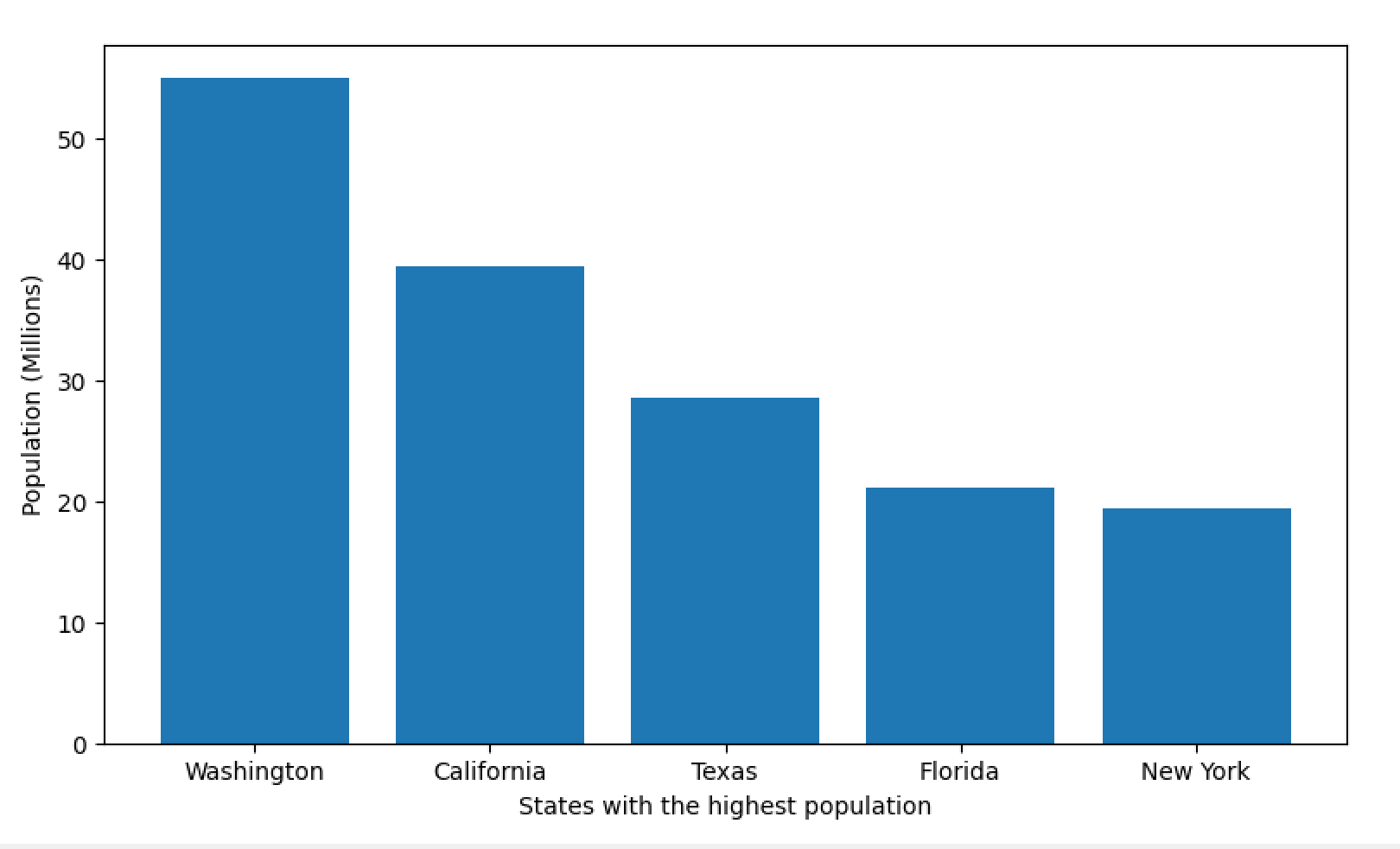
Run 1:

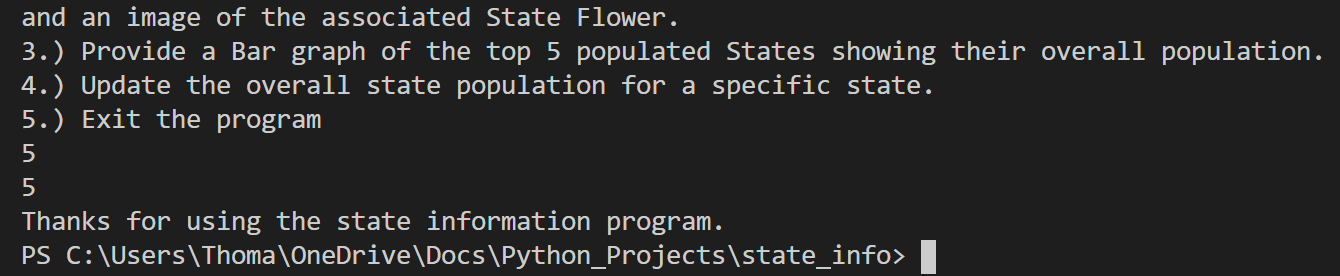




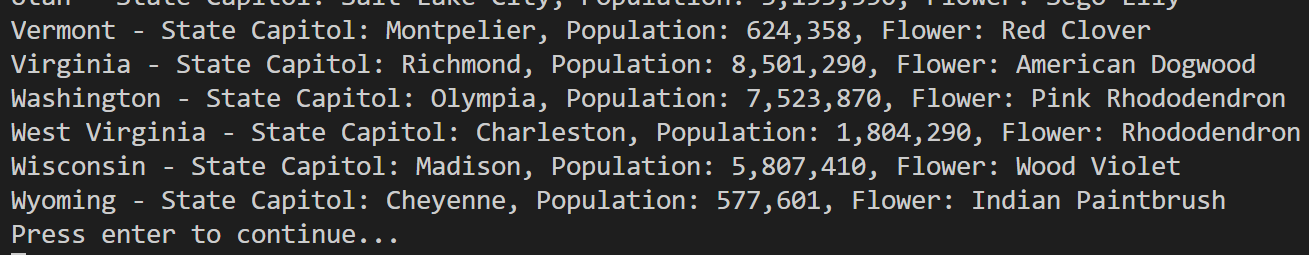


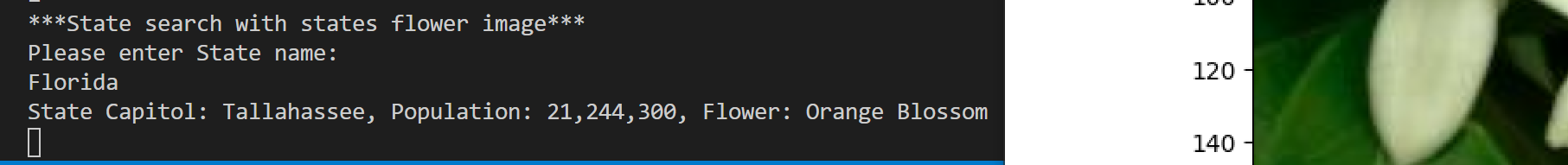


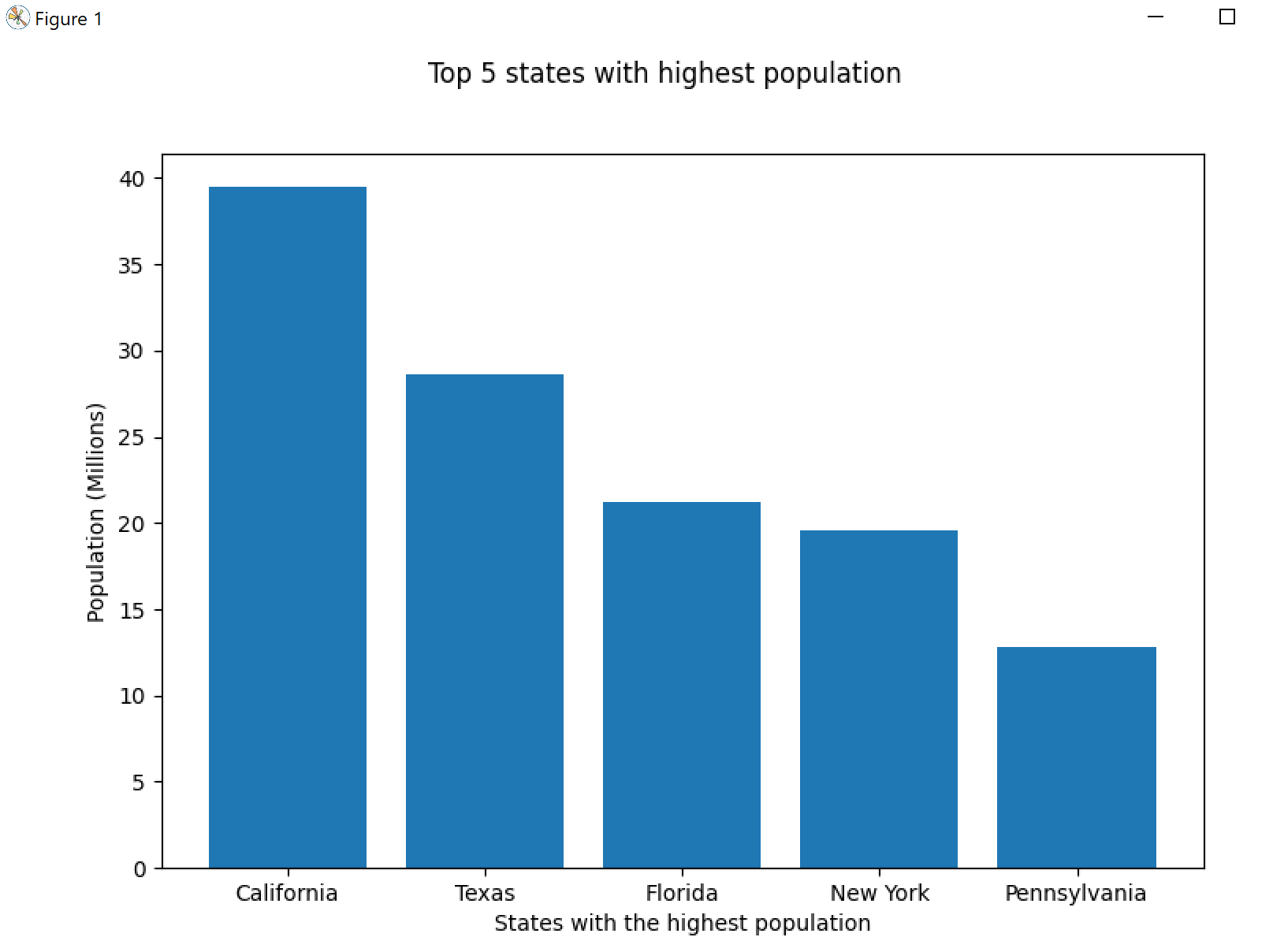


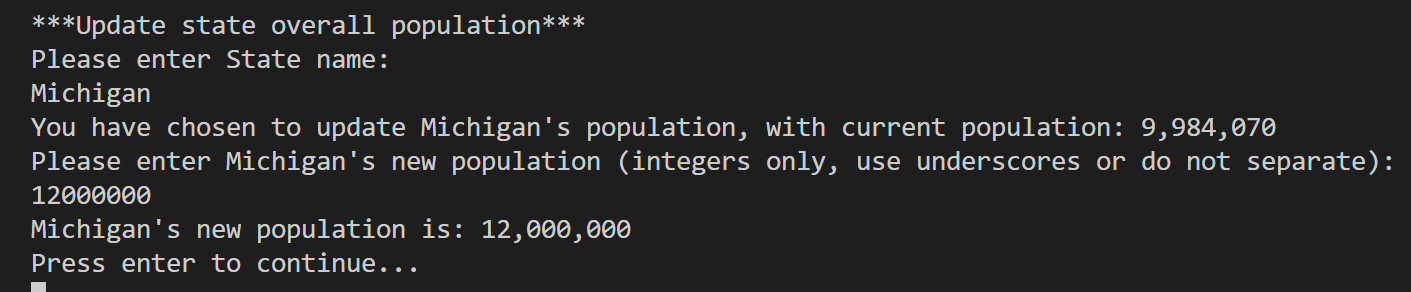


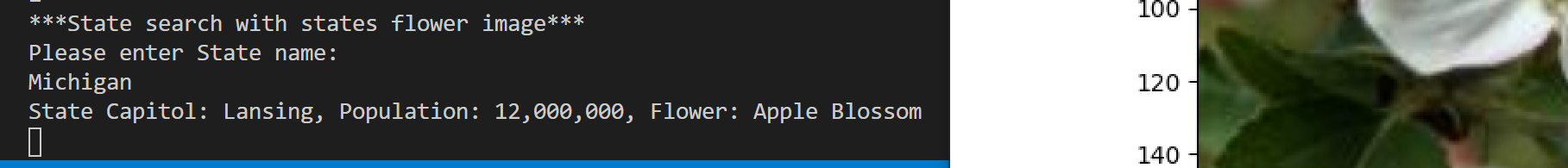
Run 2

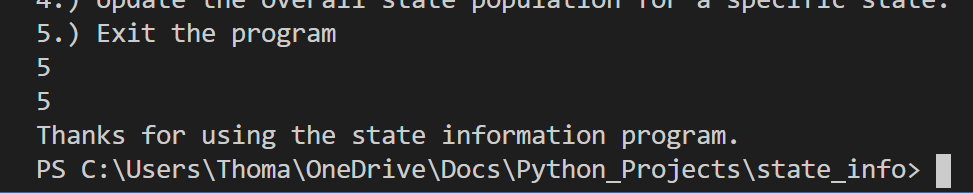




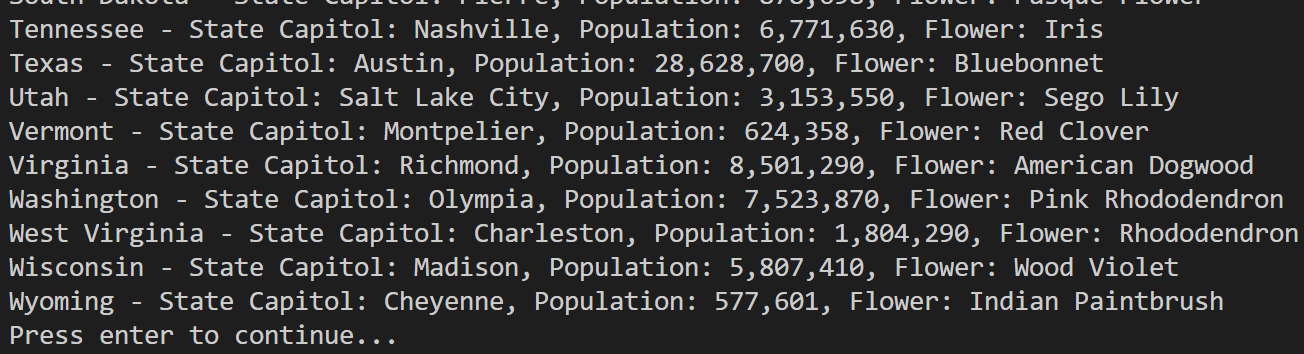


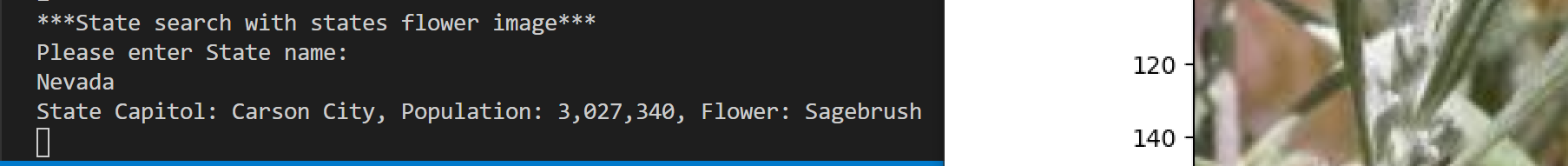


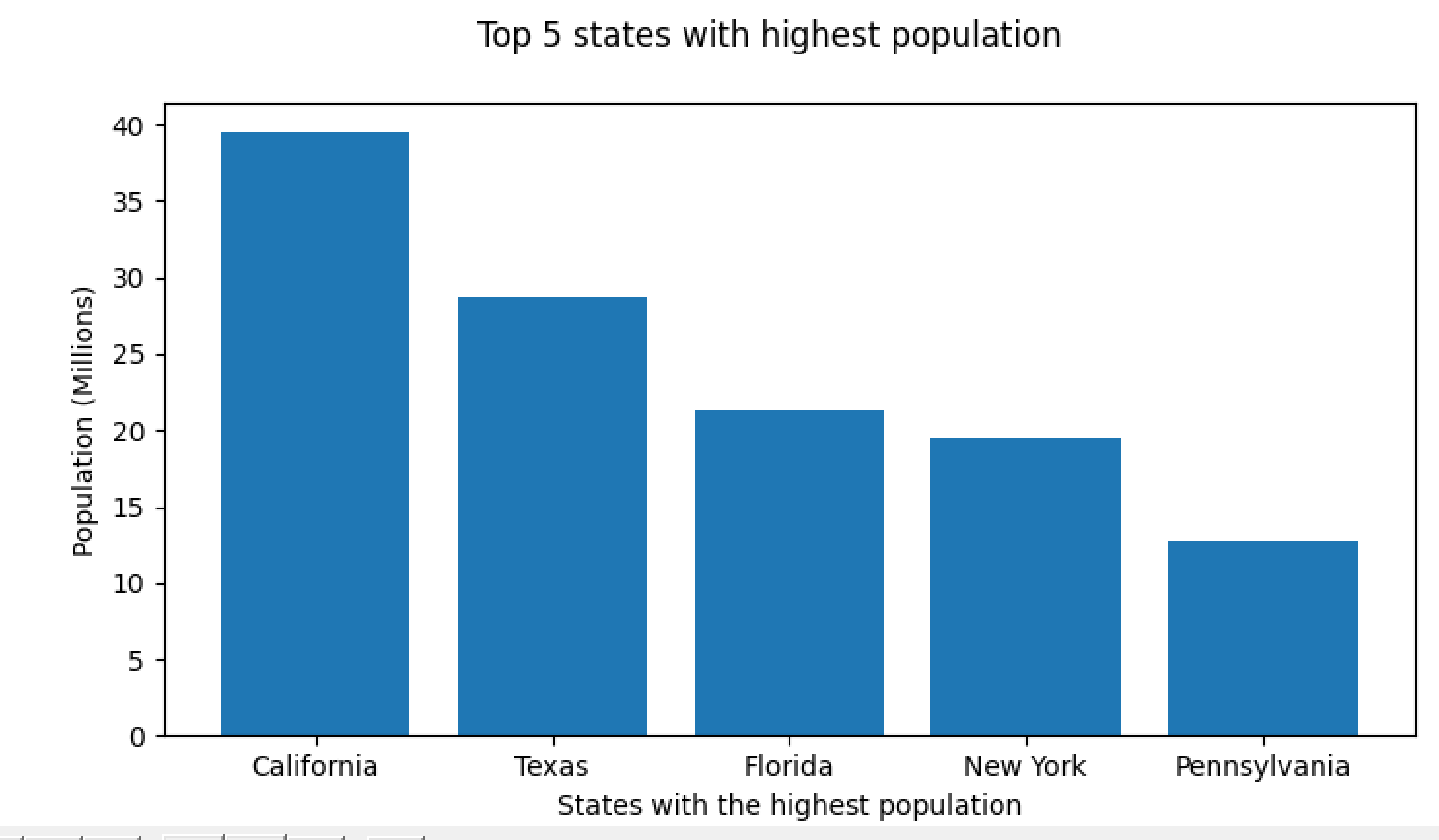


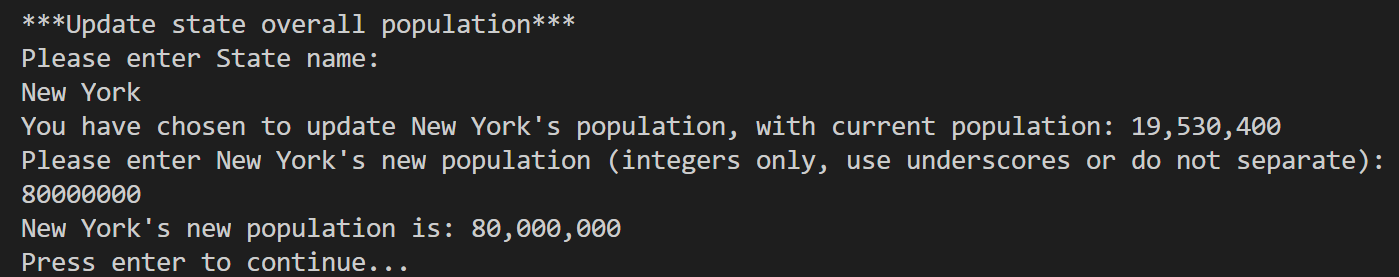


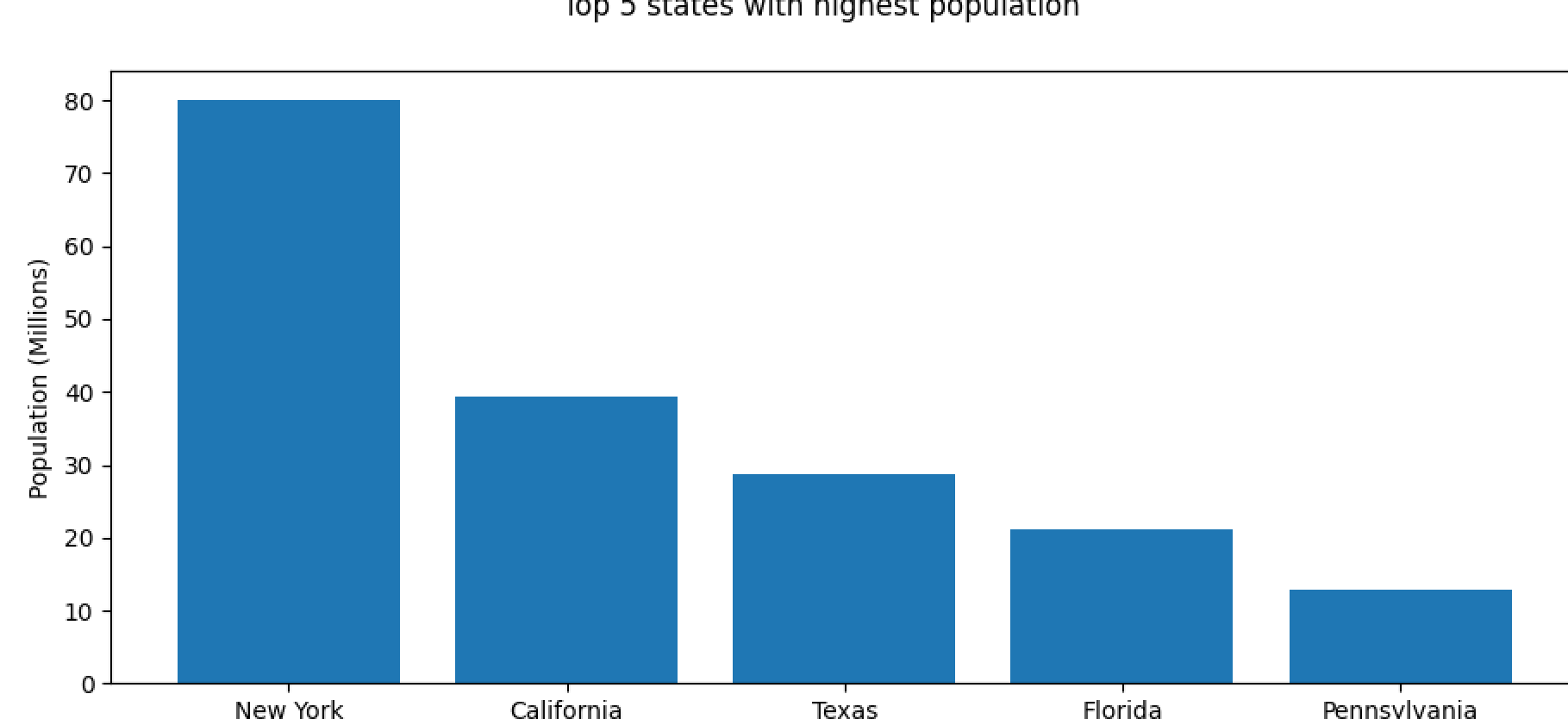
Run 3:

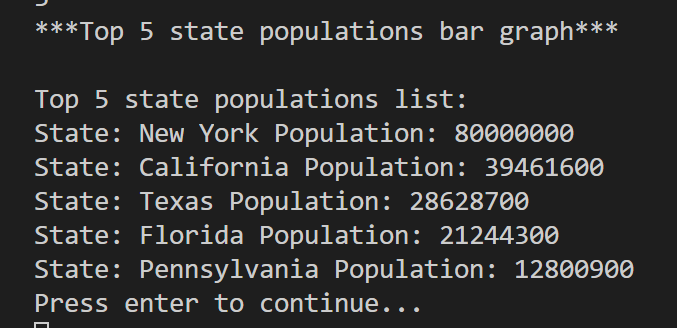










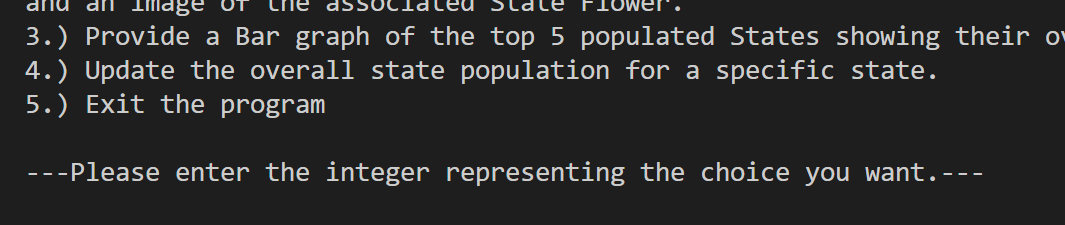


Alphabetical order test / main menu test:

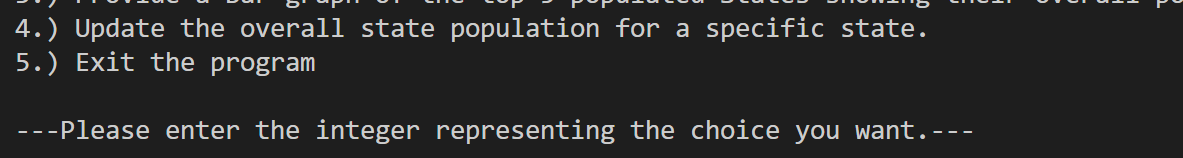
This is more of a test of what happens at the main menu when user does not make a proper selection, eventually leading to the selection of choice 1 for all states in alphabetical order

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Input | Expected output | Actual Output | Pass / Fail |
| 1 | User presses enter at main menu | Please enter integer representing the choice you want | ---Please enter the integer representing the choice you want.--- | Pass |
| 2 | User presses space then enter at main menu | Please enter integer representing the choice you want | ---Please enter the integer representing the choice you want.--- | Pass |
| 3 | User enters a character at main menu | Please enter integer representing the choice you want | ---Please enter the integer representing the choice you want.--- | Pass |
| 4 | User enters 1.5 | Please enter integer representing the choice you want | ---Please enter the integer representing the choice you want.--- | Pass |
| 5 | User enter 1 | All states in order | All States in order | Pass |

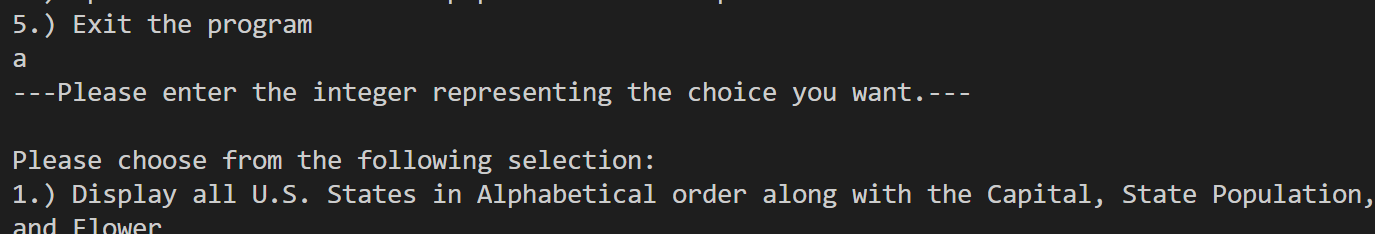
Run 1:



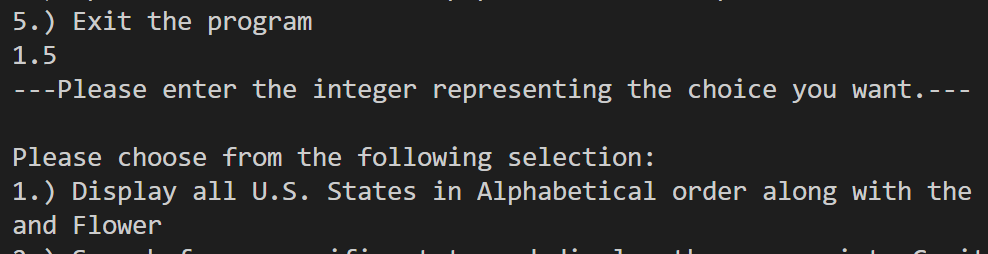
Run 2:

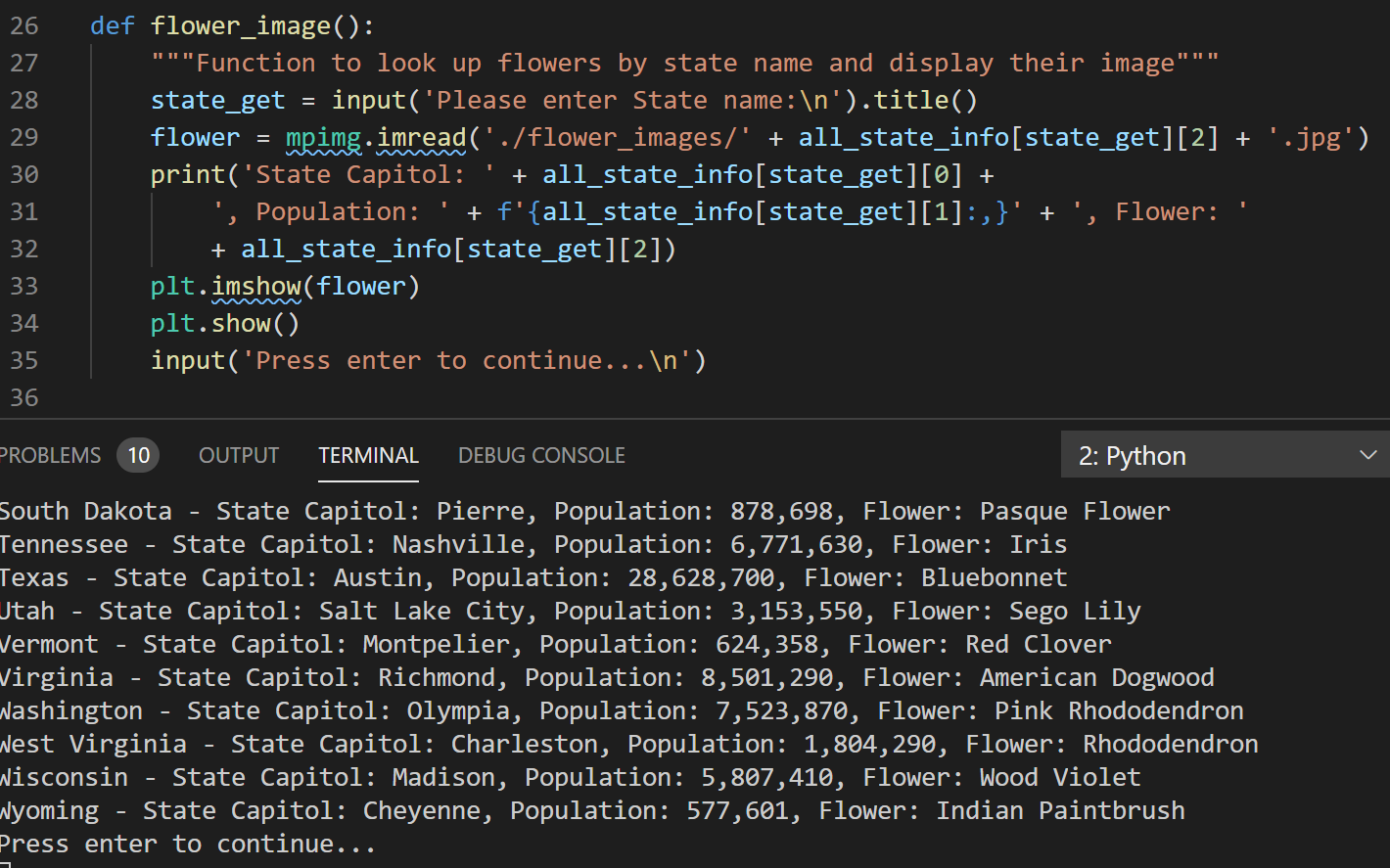


Run 3:



Run 4:



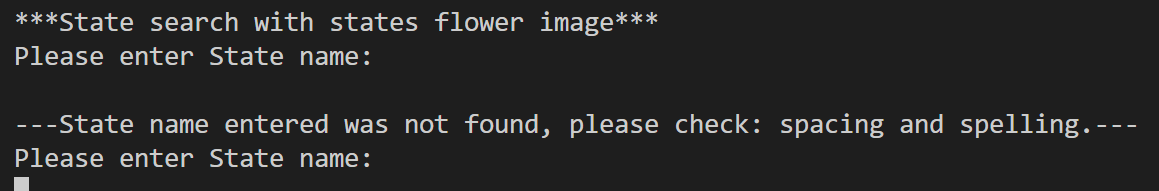
Run 5: 

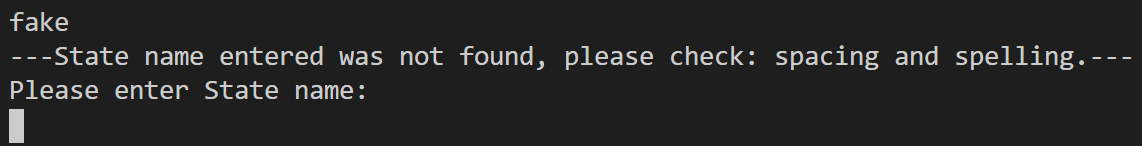
State Search Test:

Ensure program handles user input errors correctly by asking to renter

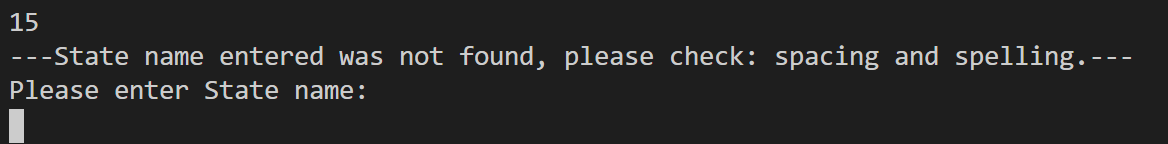
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Input | Expected Output | Output | Pass / Fail |
| 1 | User presses enter | State was not found | ---State name entered was not found, please check: spacing and spelling.--- | Pass |
| 2 | fake | State was not found | ---State name entered was not found, please check: spacing and spelling.--- | Pass |
| 3 | 15 | State was not found | ---State name entered was not found, please check: spacing and spelling.--- | Pass |
| 4 | north dakota | States info with flower picture | States info with flower picture | Pass |
| 5 | Texas | States info with flower picture | States info with flower picture | Pass |

Run 1:

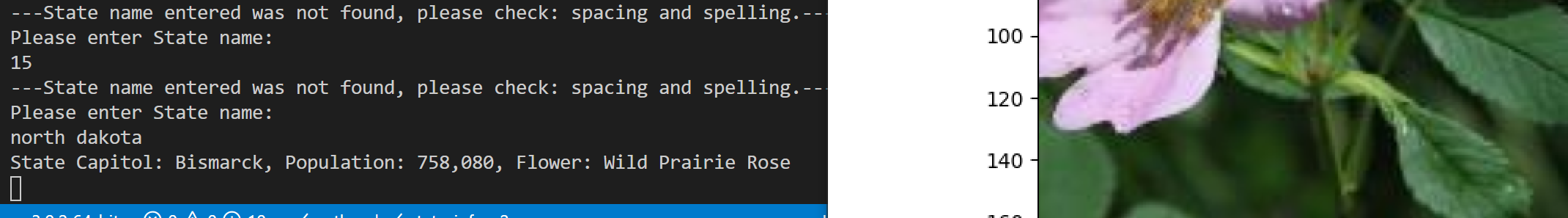


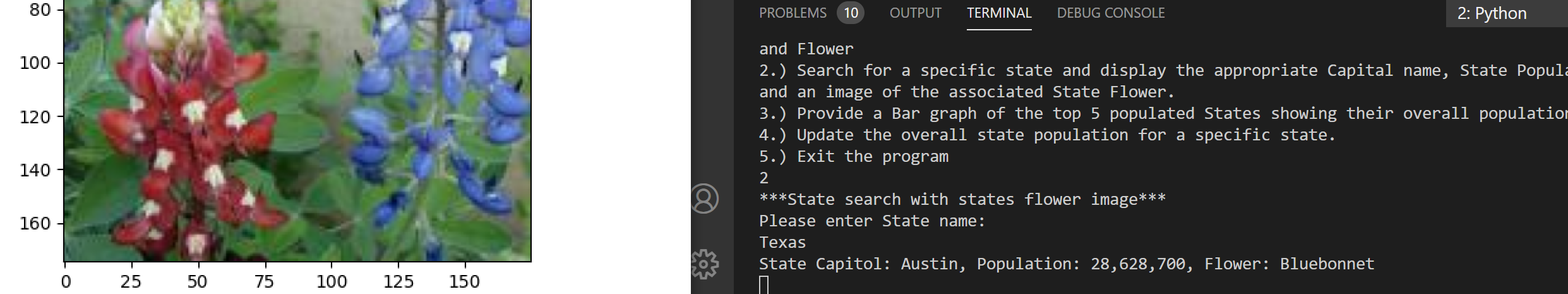
Run 2: 

Run 3:



Run 4:



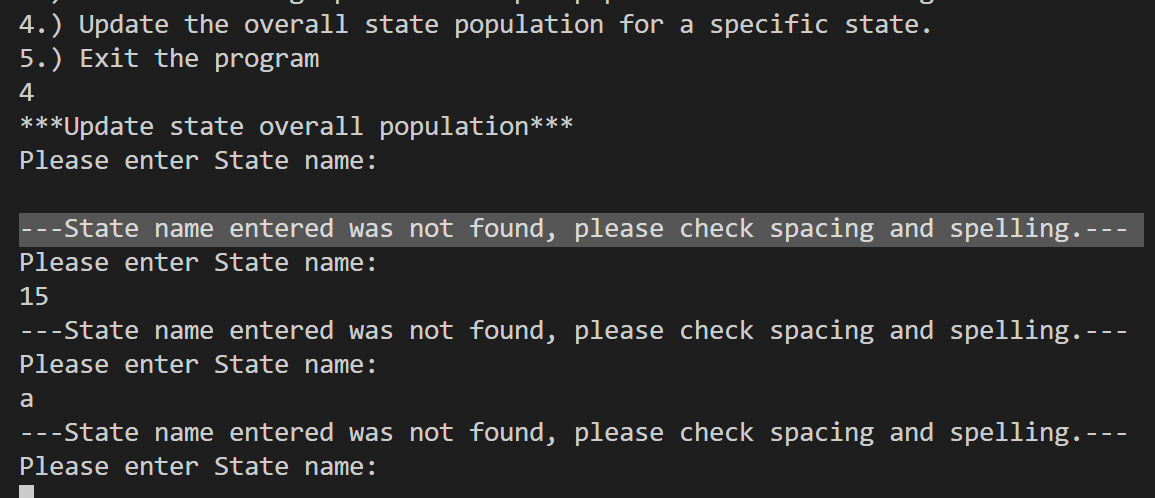
Run 5: 

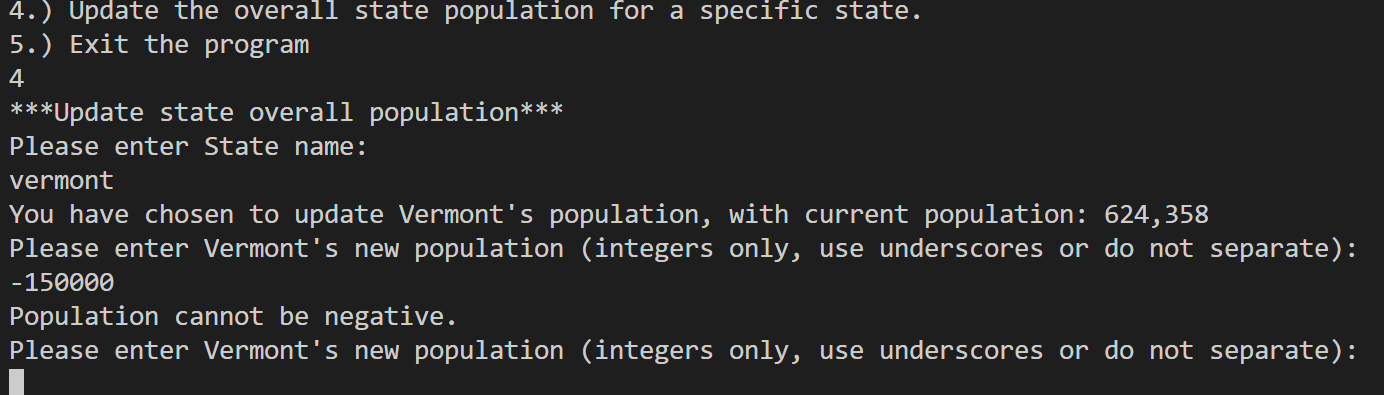
Top 5 Population and Population Update Test:

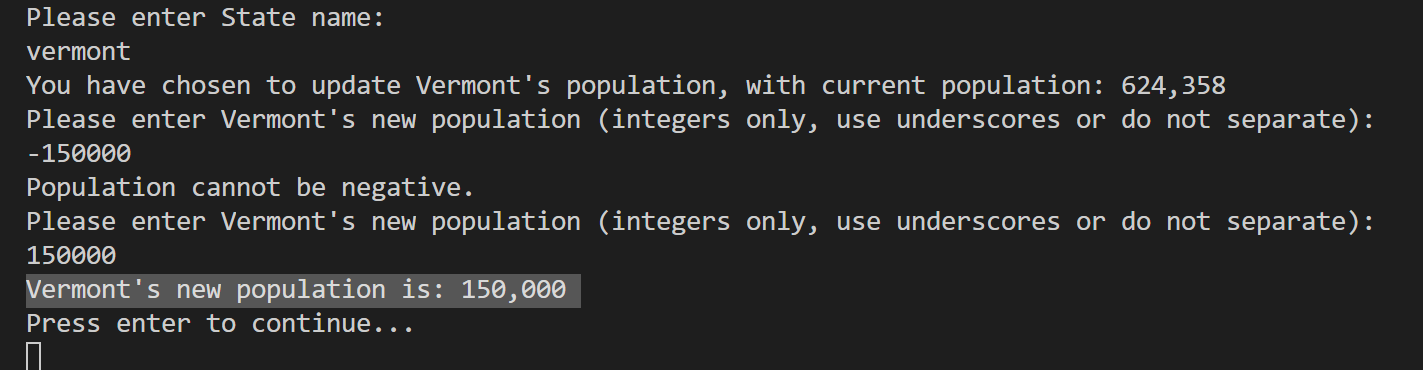
Testing together, because populations being updated should reflect on bar graph, if they have a larger population, if using a small population that will not make the graph will use state search to show that the population was updated

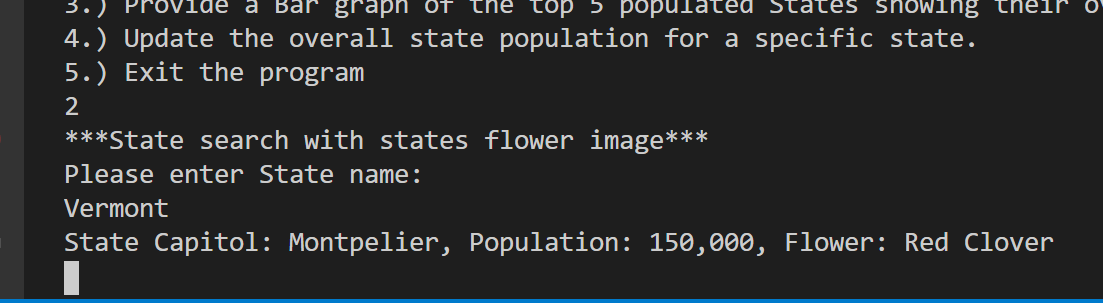
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test | Input | Expected Output | Output | Pass / Fail |
| 1 | User presses enter, then 15, then a | State name entered was not found | ---State name entered was not found, please check spacing and spelling.--- | Pass |
| 2 | Vermont, -150000 | Population cannot be negative | Population cannot be negative.  Please enter Vermont's new population (integers only, use underscores or do not separate): | Pass |
| 3 | 150000 | Vermont’s new population is 150000 | Vermont's new population is: 150,000 | Pass |
| 4 | Virginia, 55000000 | Virginia’s new population is 55000000 | Virginia’s new population is 55000000 | Pass |
| 5 | Idaho, eight, then recover by entering correct population | New population was not accepted | ---New population was not accepted, please enter an integer with underscores or no seperators.--- | Pass |

Run 1:

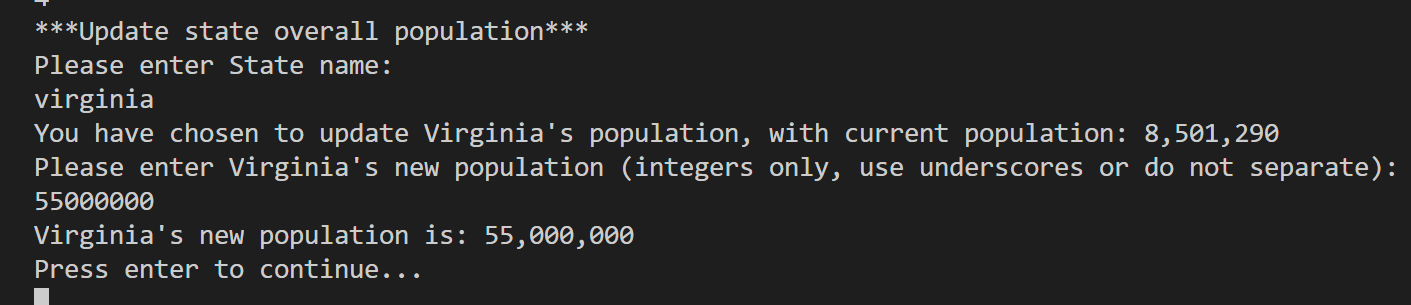


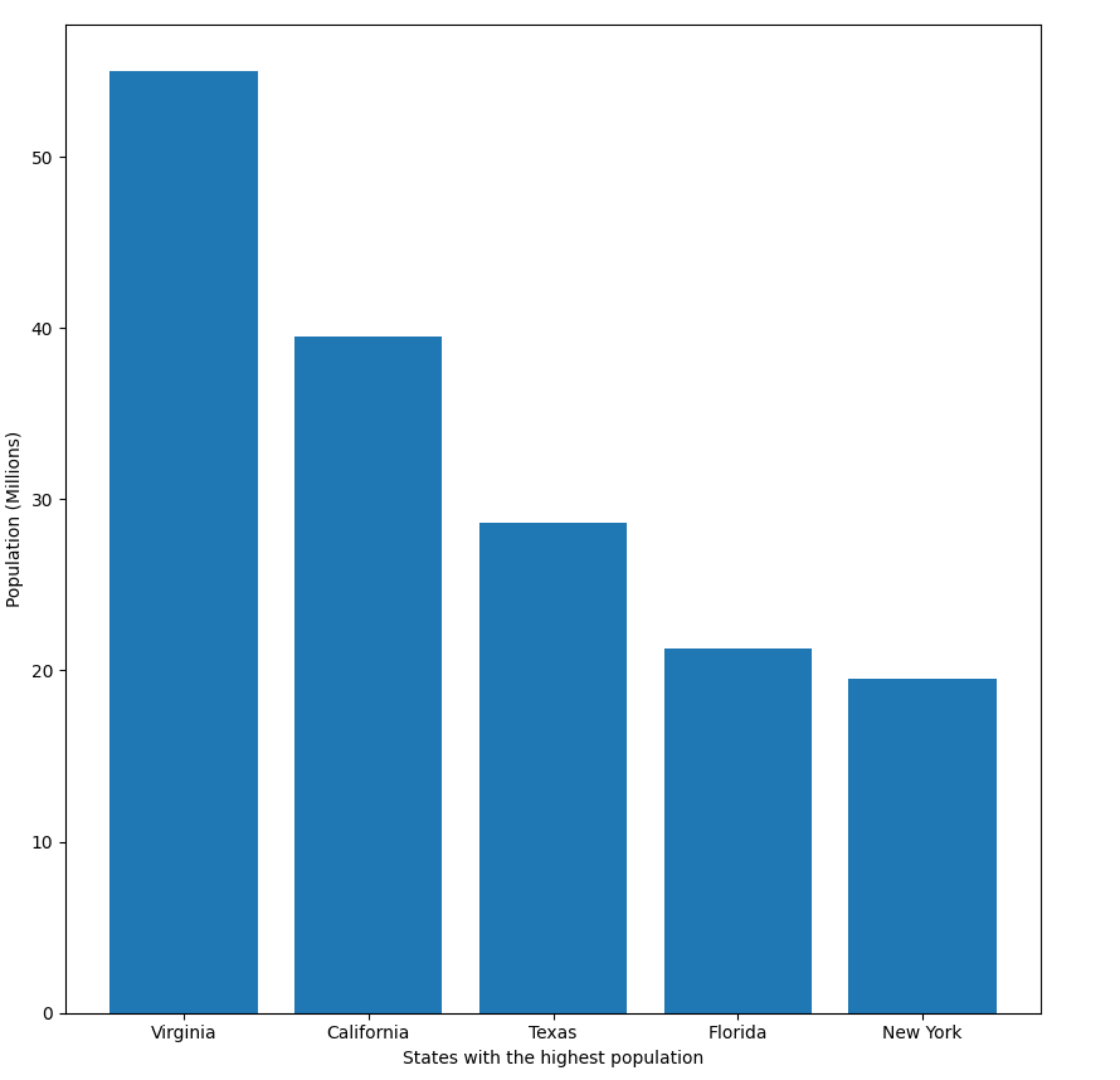
Run 2: 

Run 3: 



Run 4:





Run 5: 