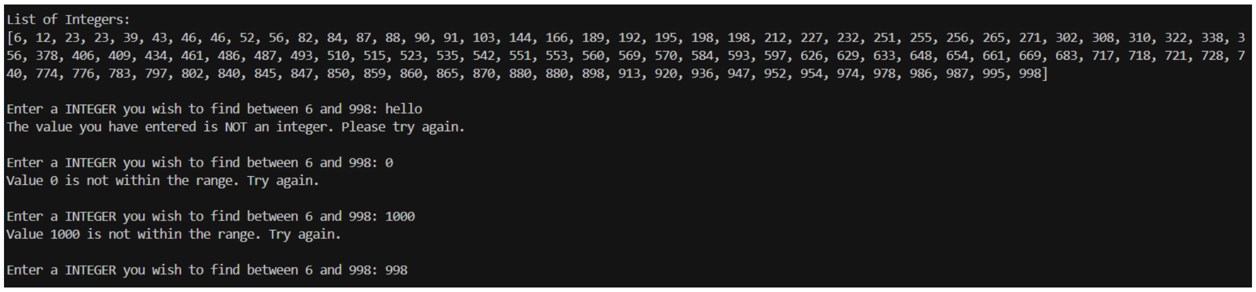
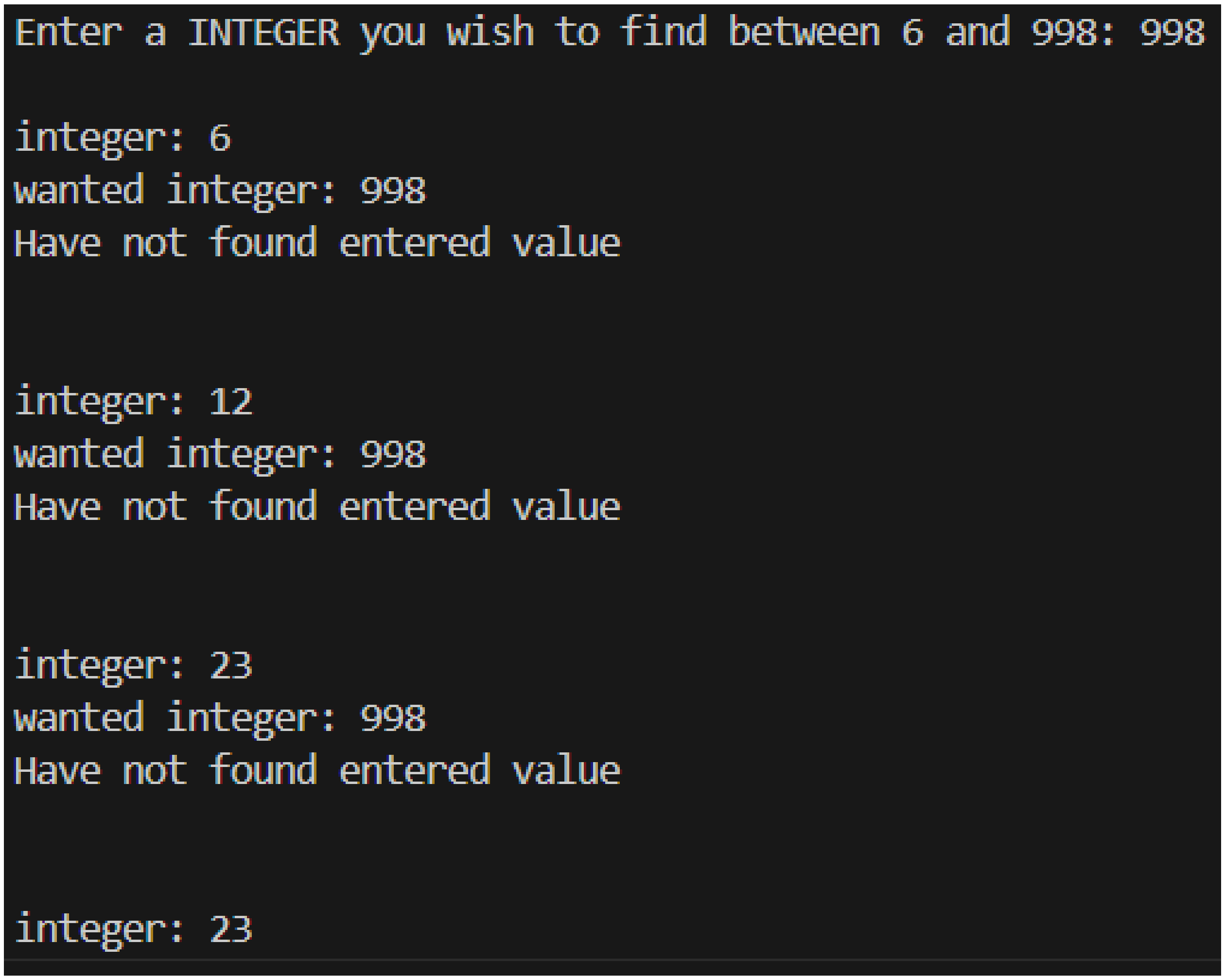
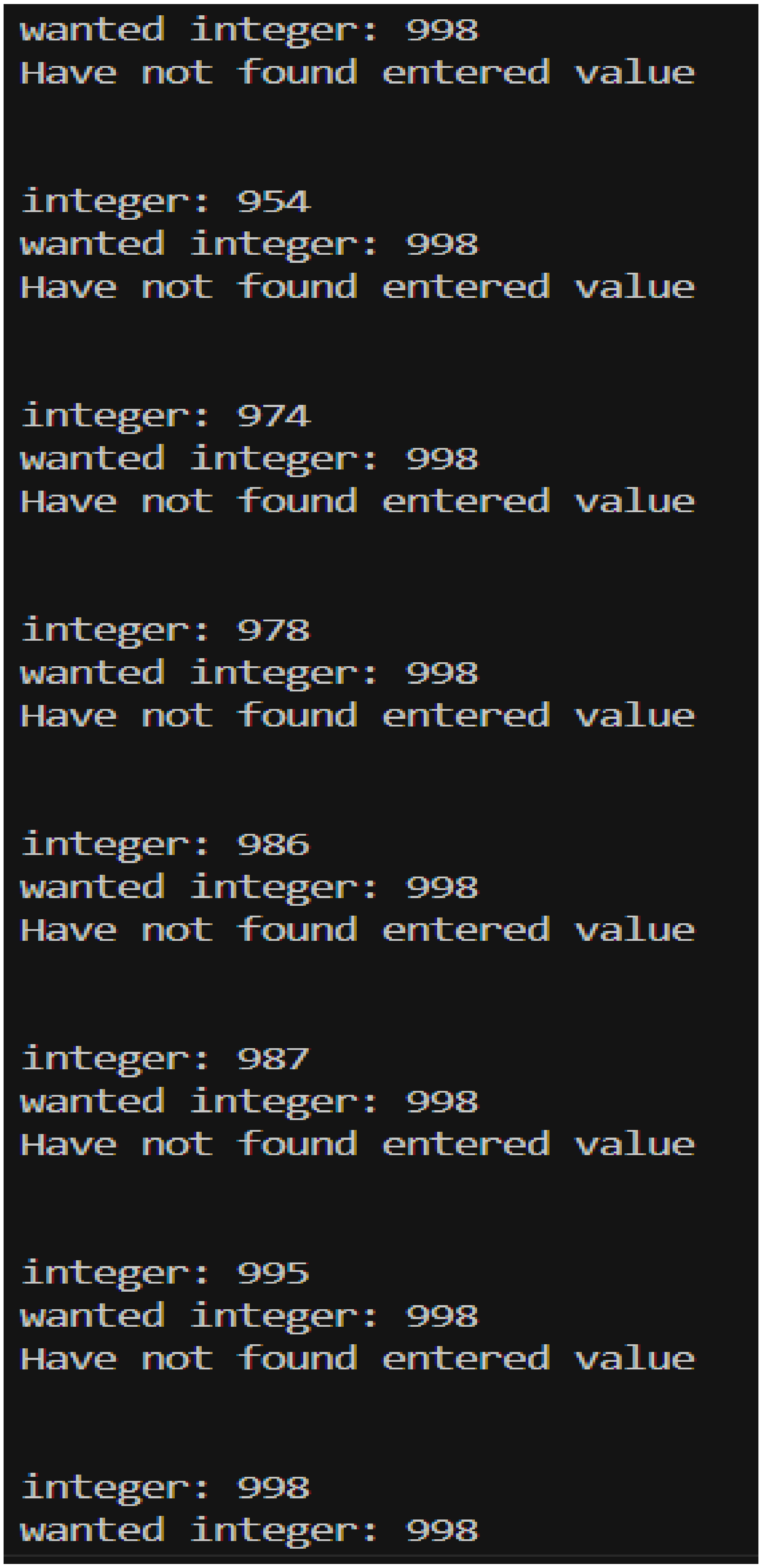
Question #1:



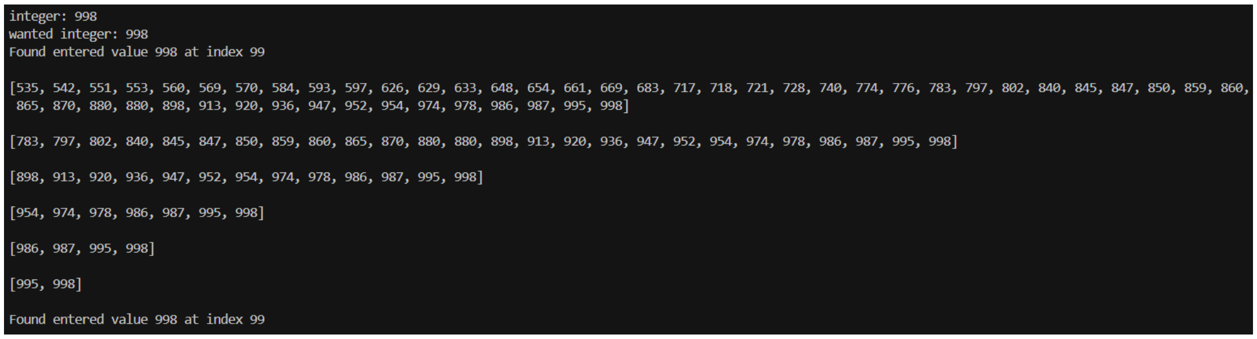
```
List of Integers:
[6, 12, 23, 23, 39, 43, 46, 46, 52, 56, 82, 84, 87, 88, 90, 91, 103, 144, 166, 189, 192, 195, 198, 198, 212, 227, 232, 251, 255, 256, 265, 271, 302, 308, 310, 322, 338, 3
56, 378, 406, 409, 434, 461, 486, 487, 493, 510, 515, 523, 535, 542, 551, 553, 560, 569, 570, 584, 593, 597, 626, 629, 633, 648, 654, 661, 669, 683, 717, 718, 721, 728, 7
40, 774, 776, 783, 797, 802, 840, 845, 847, 850, 859, 860, 865, 870, 880, 880, 898, 913, 920, 936, 947, 952, 954, 974, 978, 986, 987, 995, 998]

Enter a INTEGER you wish to find between 6 and 998: hello
The value you have entered is NOT an integer. Please try again.

Enter a INTEGER you wish to find between 6 and 998: 0
Value 0 is not within the range. Try again.

Enter a INTEGER you wish to find between 6 and 998: 1000
Value 1000 is not within the range. Try again.

Enter a INTEGER you wish to find between 6 and 998: 998
```



```
Enter a INTEGER you wish to find between 6 and 998: 998

integer: 6
wanted integer: 998
Have not found entered value



integer: 12
wanted integer: 998
Have not found entered value



integer: 23
wanted integer: 998
Have not found entered value



integer: 23
```



```
wanted integer: 998
Have not found entered value


integer: 954
wanted integer: 998
Have not found entered value


integer: 974
wanted integer: 998
Have not found entered value


integer: 978
wanted integer: 998
Have not found entered value


integer: 986
wanted integer: 998
Have not found entered value


integer: 987
wanted integer: 998
Have not found entered value


integer: 995
wanted integer: 998
Have not found entered value


integer: 998
wanted integer: 998
```



```
integer: 998
wanted integer: 998
Found entered value 998 at index 99

[535, 542, 551, 553, 560, 569, 570, 584, 593, 597, 626, 629, 633, 648, 654, 661, 669, 683, 717, 718, 721, 728, 740, 774, 776, 783, 797, 802, 840, 845, 847, 850, 859, 860,
 865, 870, 880, 880, 898, 913, 920, 936, 947, 952, 954, 974, 978, 986, 987, 995, 998]

[783, 797, 802, 840, 845, 847, 850, 859, 860, 865, 870, 880, 880, 898, 913, 920, 936, 947, 952, 954, 974, 978, 986, 987, 995, 998]

[898, 913, 920, 936, 947, 952, 954, 974, 978, 986, 987, 995, 998]

[954, 974, 978, 986, 987, 995, 998]

[986, 987, 995, 998]

[995, 998]

Found entered value 998 at index 99
```

For question #1, I decided to not implement the GUI, but still correctly searched the data with clear representation within the terminal. The initial list is displayed, along with each linear search iteration, and the searched value with the current iterated element. When it comes to the binary search, the searched array is displayed, along with every time its size is cut in half, to show its difference compared to the linear searching algorithm.

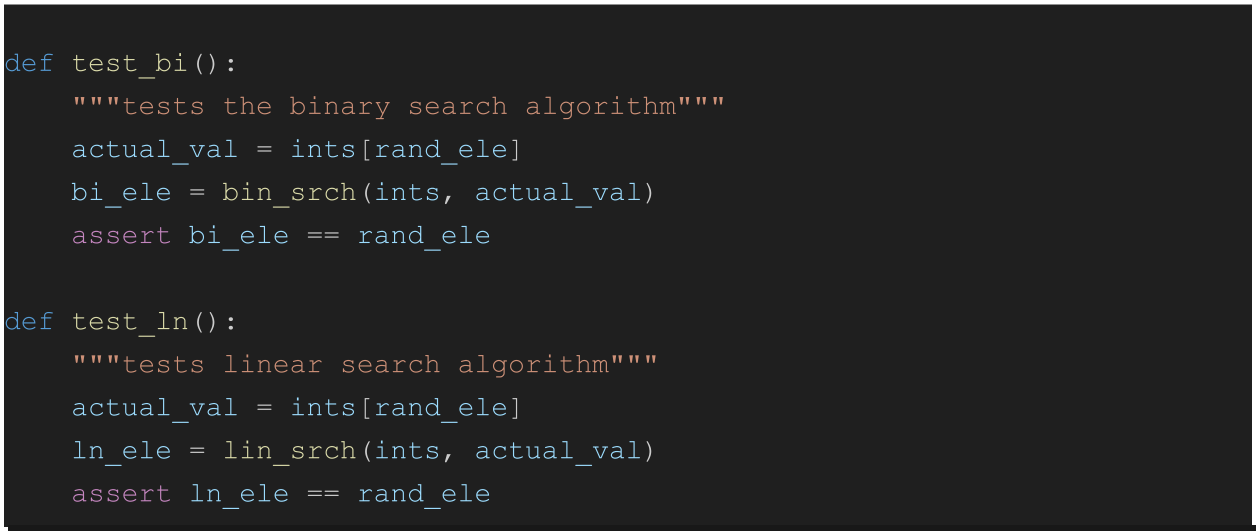
Question #2:

Some tests that would be useful for this program are making sure that the computed indices of the element within the list are correct. To do this, I would just need to make sure that the computed indices gave the integer that I was looking for.

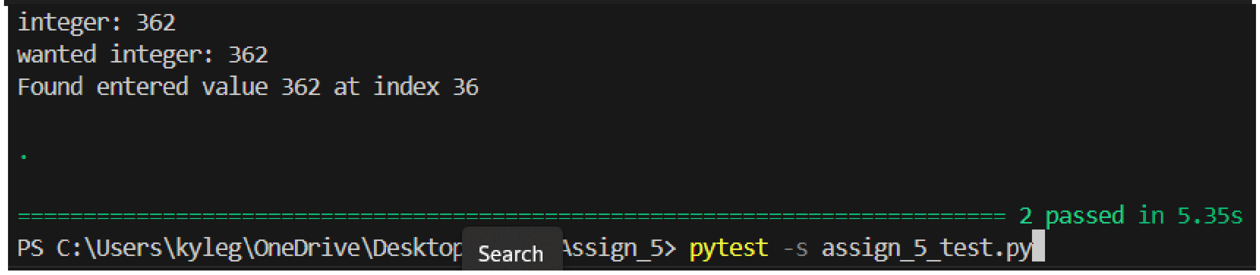
Question #3:

To implement this, I created two tests, one for the binary searching algorithm and the linear searching algorithm. To avoid required input data, I generated random integers to find within the array that I knew were located there. This was accomplished by randomly generating indices for the list and selecting an element from the list.

```python
from assign_5 import lin_srch, bin_srch
import random

low = 0
high = 1000
num_int = 100
cut_off = 1000
ints = []

for i in range(num_int):
    ints.append(random.randint(low,high))
ints.sort()

rand_ele = random.randint(0,100)
```



```python
def test_bi():
    """tests the binary search algorithm"""
    actual_val = ints[rand_ele]
    bi_ele = bin_srch(ints, actual_val)
    assert bi_ele == rand_ele

def test_ln():
    """tests linear search algorithm"""
    actual_val = ints[rand_ele]
    ln_ele = lin_srch(ints, actual_val)
    assert ln_ele == rand_ele
```



```
integer: 362
wanted integer: 362
Found entered value 362 at index 36

.

=========================================================================== 2 passed in 5.35s
PS C:\Users\kyleg\OneDrive\Desktop Search Assign_5> pytest -s assign_5_test.py
```