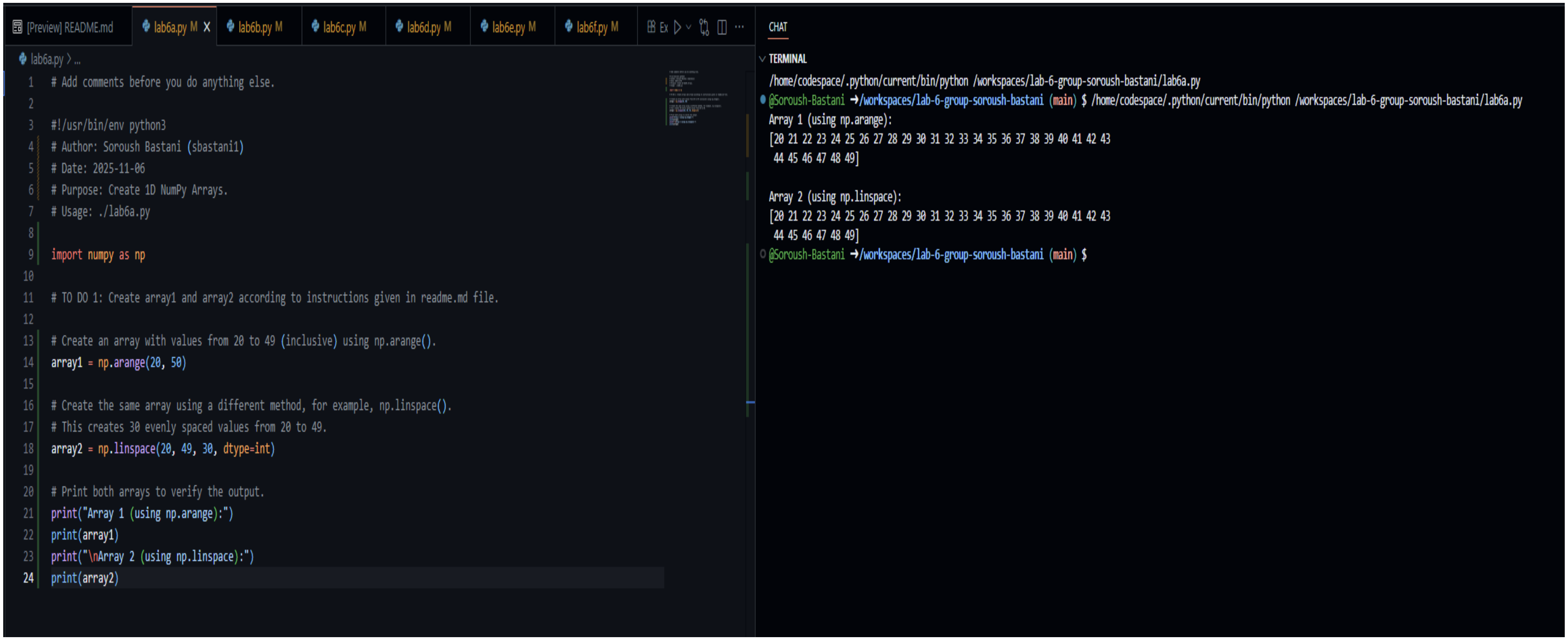
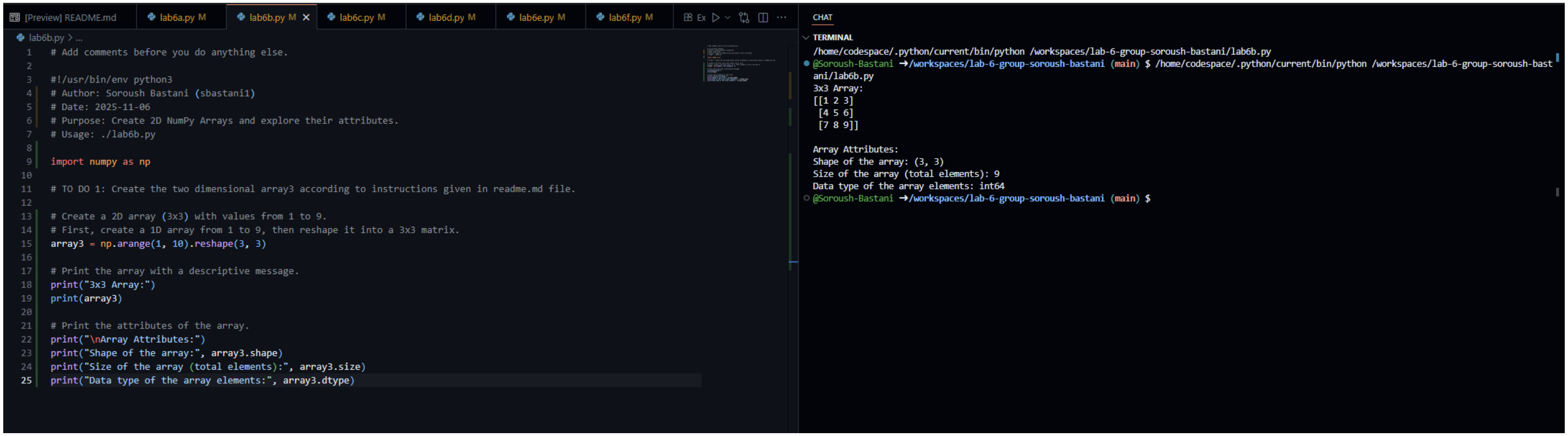
lab6a



```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Create 1D NumPy Arrays.
# Usage: ./lab6a.py

import numpy as np

# TO DO 1: Create array1 and array2 according to instructions given in readme.md file.

# Create an array with values from 20 to 49 (inclusive) using np.arange().
array1 = np.arange(20, 50)

# Create the same array using a different method, for example, np.linspace().
# This creates 30 evenly spaced values from 20 to 49.
array2 = np.linspace(20, 49, 30, dtype=int)

# Print both arrays to verify the output.
print("Array 1 (using np.arange):")
print(array1)
print("\nArray 2 (using np.linspace):")
print(array2)
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6a.py
● @Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6a.py
Array 1 (using np.arange):
[20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
 44 45 46 47 48 49]

Array 2 (using np.linspace):
[20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43
 44 45 46 47 48 49]
○ @Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```

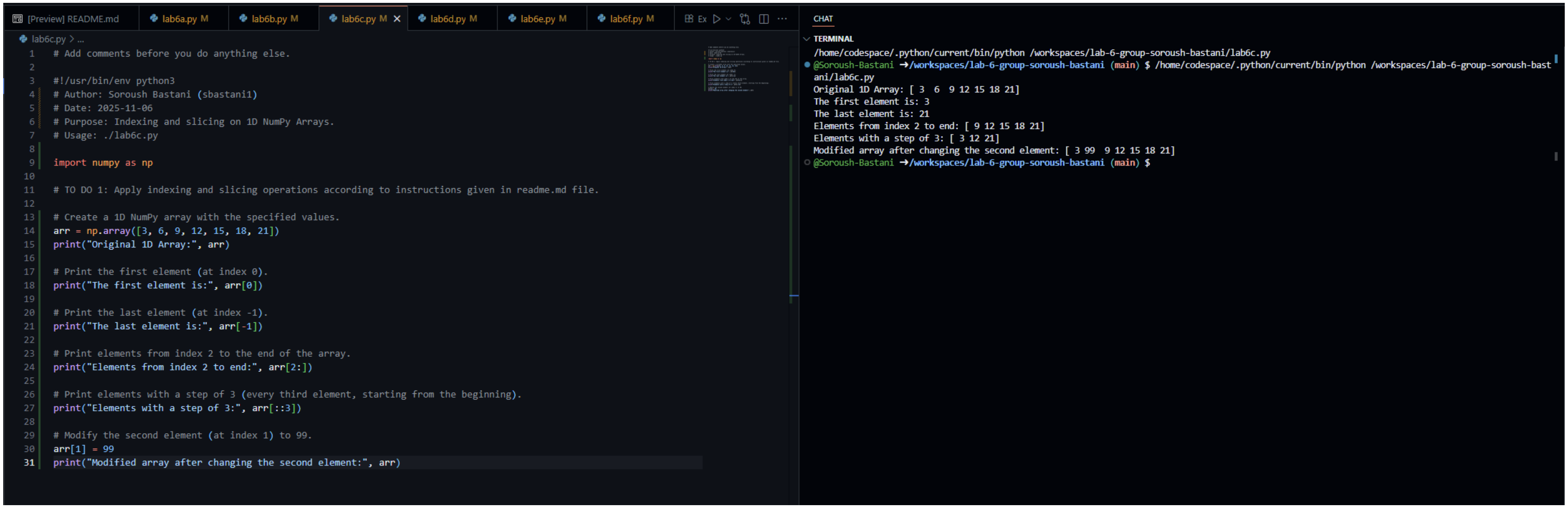
lab6b



```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Create 2D NumPy Arrays and explore their attributes.
# Usage: ./lab6b.py

import numpy as np

# TO DO 1: Create the two dimensional array3 according to instructions given in readme.md file.

# Create a 2D array (3x3) with values from 1 to 9.
# First, create a 1D array from 1 to 9, then reshape it into a 3x3 matrix.
array3 = np.arange(1, 10).reshape(3, 3)

# Print the array with a descriptive message.
print("3x3 Array:")
print(array3)

# Print the attributes of the array.
print("\nArray Attributes:")
print("Shape of the array:", array3.shape)
print("Size of the array (total elements):", array3.size)
print("Data type of the array elements:", array3.dtype)
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6b.py
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bast
ani/lab6b.py
3x3 Array:
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Array Attributes:
Shape of the array: (3, 3)
Size of the array (total elements): 9
Data type of the array elements: int64
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```

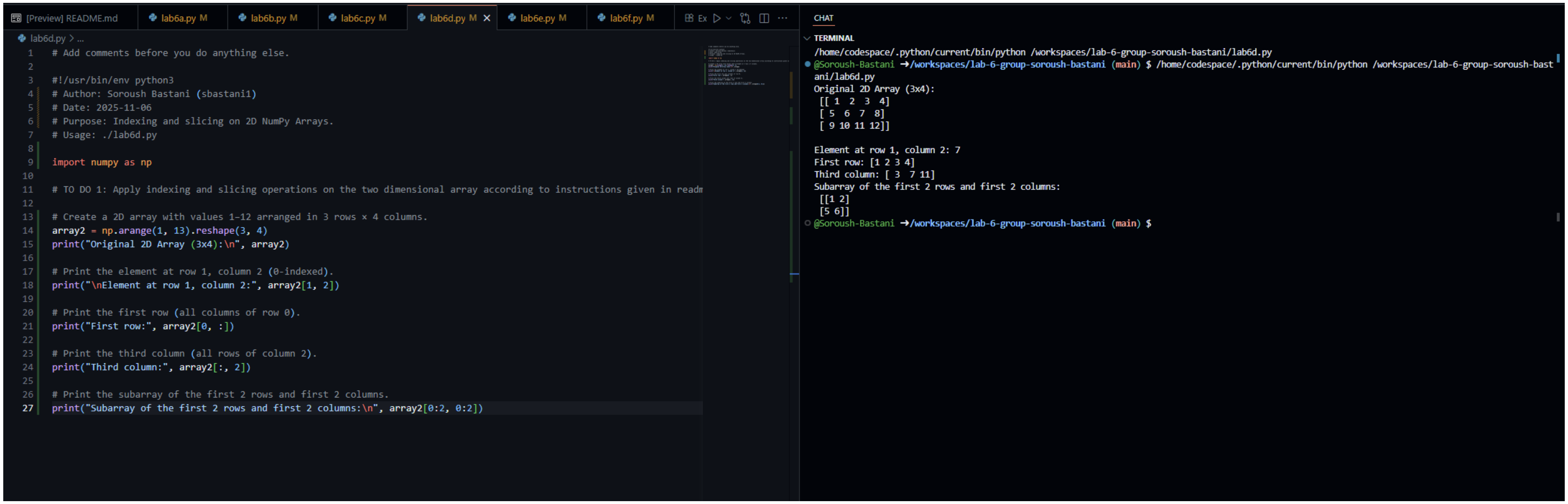
lab6c



```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Indexing and slicing on 1D NumPy Arrays.
# Usage: ./lab6c.py

import numpy as np

# TO DO 1: Apply indexing and slicing operations according to instructions given in readme.md file.

# Create a 1D NumPy array with the specified values.
arr = np.array([3, 6, 9, 12, 15, 18, 21])
print("Original 1D Array:", arr)

# Print the first element (at index 0).
print("The first element is:", arr[0])

# Print the last element (at index -1).
print("The last element is:", arr[-1])

# Print elements from index 2 to the end of the array.
print("Elements from index 2 to end:", arr[2:])

# Print elements with a step of 3 (every third element, starting from the beginning).
print("Elements with a step of 3:", arr[::3])

# Modify the second element (at index 1) to 99.
arr[1] = 99
print("Modified array after changing the second element:", arr)
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6c.py
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bast
ani/lab6c.py
Original 1D Array: [ 3  6  9 12 15 18 21]
The first element is: 3
The last element is: 21
Elements from index 2 to end: [ 9 12 15 18 21]
Elements with a step of 3: [ 3 12 21]
Modified array after changing the second element: [ 3 99  9 12 15 18 21]
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```

lab6d

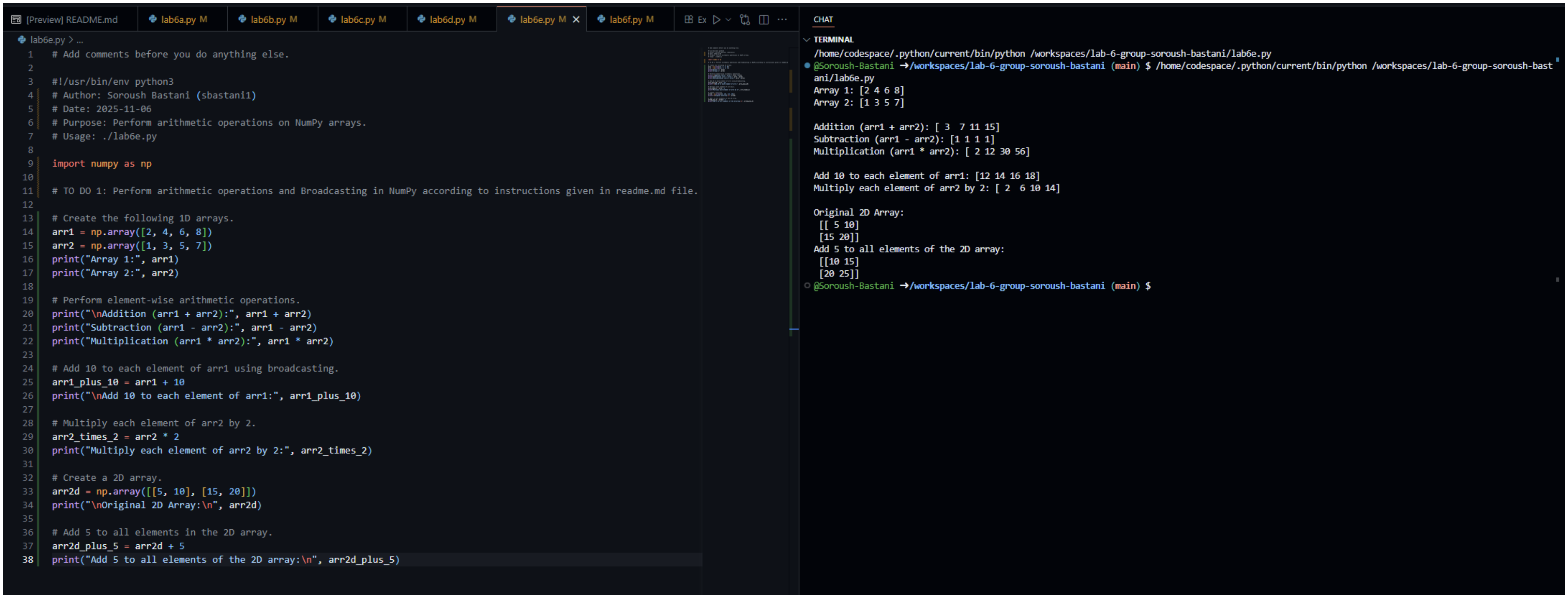


lab6d.py > ...

```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Indexing and slicing on 2D NumPy Arrays.
# Usage: ./lab6d.py

import numpy as np

# TO DO 1: Apply indexing and slicing operations on the two dimensional array according to instructions given in readm

# Create a 2D array with values 1-12 arranged in 3 rows × 4 columns.
array2 = np.arange(1, 13).reshape(3, 4)
print("Original 2D Array (3x4):\n", array2)

# Print the element at row 1, column 2 (0-indexed).
print("\nElement at row 1, column 2:", array2[1, 2])

# Print the first row (all columns of row 0).
print("First row:", array2[0, :])

# Print the third column (all rows of column 2).
print("Third column:", array2[:, 2])

# Print the subarray of the first 2 rows and first 2 columns.
print("Subarray of the first 2 rows and first 2 columns:\n", array2[0:2, 0:2])
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6d.py
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bast
ani/lab6d.py
Original 2D Array (3x4):
 [[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]

Element at row 1, column 2: 7
First row: [1 2 3 4]
Third column: [ 3  7 11]
Subarray of the first 2 rows and first 2 columns:
 [[1 2]
 [5 6]]
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```

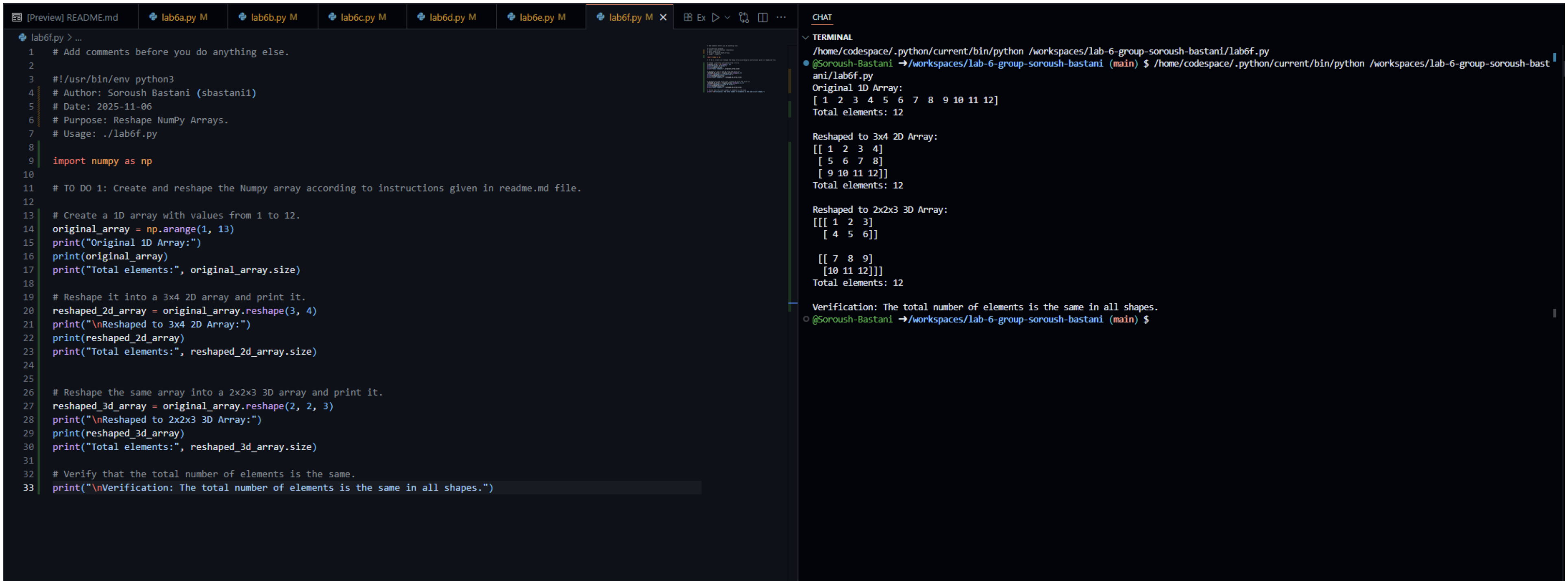
lab6e



```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Perform arithmetic operations on NumPy arrays.
# Usage: ./lab6e.py

import numpy as np

# TO DO 1: Perform arithmetic operations and Broadcasting in NumPy according to instructions given in readme.md file.

# Create the following 1D arrays.
arr1 = np.array([2, 4, 6, 8])
arr2 = np.array([1, 3, 5, 7])
print("Array 1:", arr1)
print("Array 2:", arr2)

# Perform element-wise arithmetic operations.
print("\nAddition (arr1 + arr2):", arr1 + arr2)
print("Subtraction (arr1 - arr2):", arr1 - arr2)
print("Multiplication (arr1 * arr2):", arr1 * arr2)

# Add 10 to each element of arr1 using broadcasting.
arr1_plus_10 = arr1 + 10
print("\nAdd 10 to each element of arr1:", arr1_plus_10)

# Multiply each element of arr2 by 2.
arr2_times_2 = arr2 * 2
print("Multiply each element of arr2 by 2:", arr2_times_2)

# Create a 2D array.
arr2d = np.array([[5, 10], [15, 20]])
print("\nOriginal 2D Array:\n", arr2d)

# Add 5 to all elements in the 2D array.
arr2d_plus_5 = arr2d + 5
print("Add 5 to all elements of the 2D array:\n", arr2d_plus_5)
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6e.py
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bast
ani/lab6e.py
Array 1: [2 4 6 8]
Array 2: [1 3 5 7]

Addition (arr1 + arr2): [ 3  7 11 15]
Subtraction (arr1 - arr2): [1 1 1 1]
Multiplication (arr1 * arr2): [ 2 12 30 56]

Add 10 to each element of arr1: [12 14 16 18]
Multiply each element of arr2 by 2: [ 2  6 10 14]

Original 2D Array:
 [[ 5 10]
 [15 20]]
Add 5 to all elements of the 2D array:
 [[10 15]
 [20 25]]
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```

lab6f



```python
# Add comments before you do anything else.

#!/usr/bin/env python3
# Author: Soroush Bastani (sbastani1)
# Date: 2025-11-06
# Purpose: Reshape NumPy Arrays.
# Usage: ./lab6f.py

import numpy as np

# TO DO 1: Create and reshape the Numpy array according to instructions given in readme.md file.

# Create a 1D array with values from 1 to 12.
original_array = np.arange(1, 13)
print("Original 1D Array:")
print(original_array)
print("Total elements:", original_array.size)

# Reshape it into a 3×4 2D array and print it.
reshaped_2d_array = original_array.reshape(3, 4)
print("\nReshaped to 3x4 2D Array:")
print(reshaped_2d_array)
print("Total elements:", reshaped_2d_array.size)


# Reshape the same array into a 2×2×3 3D array and print it.
reshaped_3d_array = original_array.reshape(2, 2, 3)
print("\nReshaped to 2x2x3 3D Array:")
print(reshaped_3d_array)
print("Total elements:", reshaped_3d_array.size)

# Verify that the total number of elements is the same.
print("\nVerification: The total number of elements is the same in all shapes.")
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bastani/lab6f.py
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $ /home/codespace/.python/current/bin/python /workspaces/lab-6-group-soroush-bast
ani/lab6f.py
Original 1D Array:
[ 1  2  3  4  5  6  7  8  9 10 11 12]
Total elements: 12

Reshaped to 3x4 2D Array:
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
Total elements: 12

Reshaped to 2x2x3 3D Array:
[[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [10 11 12]]]
Total elements: 12

Verification: The total number of elements is the same in all shapes.
@Soroush-Bastani ➜/workspaces/lab-6-group-soroush-bastani (main) $
```