

Lab4a



The screenshot shows the VS Code interface with the following details:

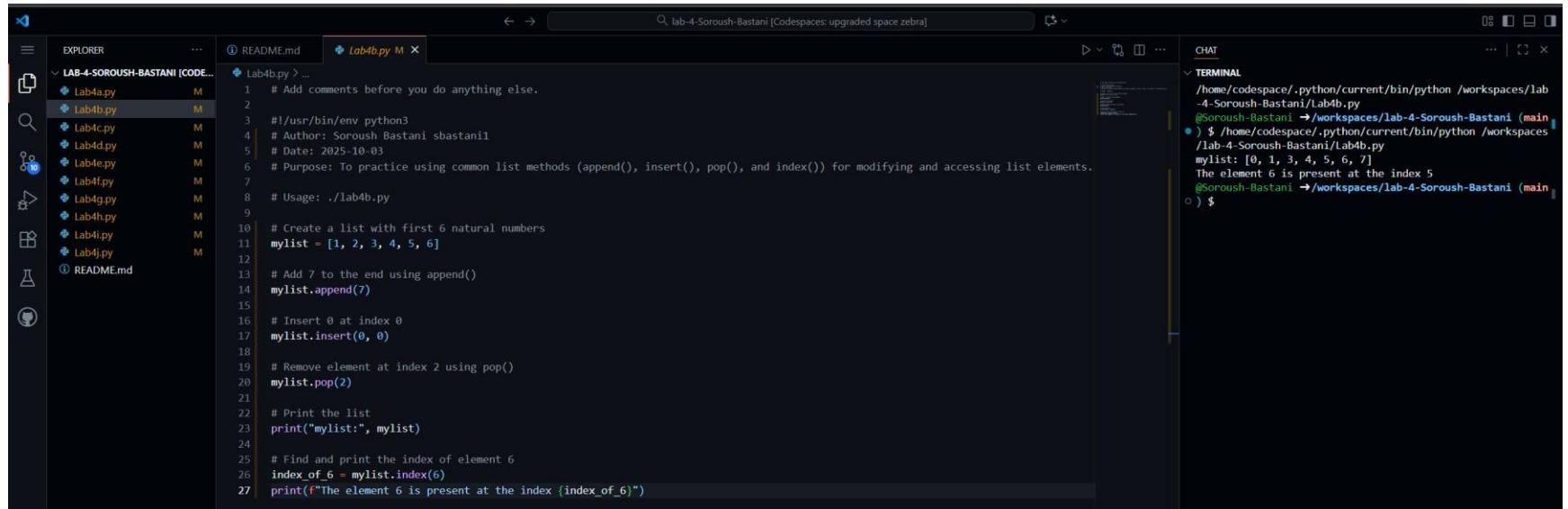
- EXPLORER:** Shows a folder named "LAB-4-SOROUSH-BASTANI (CODE...)" containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py, Lab4j.py, and README.md.
- EDITOR:** The active file is "Lab4a.py". The code content is as follows:

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-03
6 # Purpose: Create two lists and join them
7 # Usage: ./Lab4a.py
8
9 # Follow the specific instructions given in the README.md file
10 mylist1=[1,3,5]
11 mylist2=[0,2,4]
12 mylist=mylist1+mylist2
13 print(mylist),
```

TERMINAL: Shows the command-line output of running the script:

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4a.py
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
● $ /home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4a.py
[1, 3, 5, 0, 2, 4]
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
○ ) $
```

Lab4b



The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows a folder named "LAB-4-SOROUSH-BASTANI (CODE...)" containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py, Lab4j.py, and README.md.
- EDITOR:** The active file is "Lab4b.py". The code content is as follows:

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-03
6 # Purpose: To practice using common list methods (append(), insert(), pop(), and index()) for modifying and accessing list elements.
7
8 # Usage: ./Lab4b.py
9
10 # Create a list with first 6 natural numbers
11 mylist = [1, 2, 3, 4, 5, 6]
12
13 # Add 7 to the end using append()
14 mylist.append(7)
15
16 # Insert 0 at index 0
17 mylist.insert(0, 0)
18
19 # Remove element at index 2 using pop()
20 mylist.pop(2)
21
22 # Print the list
23 print("mylist:", mylist)
24
25 # Find and print the index of element 6
26 index_of_6 = mylist.index(6)
27 print(f"The element 6 is present at the index {index_of_6}")
```

TERMINAL: Shows the command-line output of running the script:

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4b.py
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
● $ /home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4b.py
mylist: [0, 1, 3, 4, 5, 6, 7]
The element 6 is present at the index 5
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
○ ) $
```

Lab4c

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder structure for "LAB-4-SOROUSH-BASTANI [CODE...]" containing files: Lab4a.py, Lab4b.py, Lab4c.py (selected), Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py, Lab4j.py, and README.md.
- EDITOR:** The file "Lab4c.py" is open, displaying the following Python code:

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-03
6 # Purpose: Create a list and update its elements.
7 # Usage: ./lab4c.py
8
9 # Follow the specific instructions given in the README.md file
10 students = ["Ama", "Eden", "Maija", "Daniel", "Ibrahim"]
11 students[1] = "Maggy"
12
13 for student in students:
14     print(student)
15
16
17
```
- TERMINAL:** A terminal window is visible on the right side of the interface, showing a session between users "Soroush-Bastani" and "Ibrahim". The session includes commands like "python /workspaces/lab-4-Soroush-Bastani/main.py" and a list of names: Ama, Maggy, Maija, Daniel, Ibrahim.

Lab4d

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder named "LAB-4-SOROUSH-BASTANI (CODE...)" containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py (selected), Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py, Lab4j.py, and README.md.
- EDITOR:** The file "Lab4d.py" is open, displaying Python code. The code defines a function `buildtheSet(divisor)` that creates a set of numbers divisible by the divisor from 0 to 50. It also defines a `main()` function that prints sets for divisors 3, 5, 7, and 11.
- TERMINAL:** The terminal window shows the execution of the script and its output. The output includes:
 - Execution of `/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4d.py`
 - Output of `s3: {0, 33, 3, 36, 6, 39, 9, 42, 12, 45, 15, 48, 18, 21, 24, 27, 30}`
 - Output of `s5: {0, 35, 5, 40, 10, 45, 15, 50, 20, 25, 30}`
 - Output of `s7: {0, 35, 7, 42, 14, 49, 21, 28}`
 - Output of `s11: {0, 33, 11, 44, 22}`

Lab4e

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder named "LAB-4-SOROU..." containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, and Lab4e.py (selected).
- EDITOR:** The file "Lab4e.py" is open, displaying Python code. The code defines a function `buildtheSet(divisor)` which creates a set of numbers from 0 to 50 that are divisible by the divisor. It also defines a function `s3_or_s5(s3, s5)` which returns elements that are in either `s3` or `s5` but not in both. The `main()` function calls these two functions and prints their results.
- TERMINAL:** The terminal window shows the execution of the script. It prints the sets `s3` and `s5`, followed by their union `s3 ^ s5`. The output is:

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Sorou.../Lab4e.py
@Sorou...-Bastani → /workspaces/lab-4-Sorou...-Bastani (main)
● ) $ /home/codespace/.python/current/bin/python /workspaces/lab-4-Sorou...-Bastani/Lab4e.py
s3: {0, 33, 3, 36, 6, 39, 9, 42, 12, 45, 15, 48, 18, 21, 24, 27, 30}
s5: {0, 35, 5, 40, 10, 45, 15, 50, 20, 25, 30}
s3_or_s5: {3, 5, 6, 9, 10, 12, 18, 20, 21, 24, 25, 27, 33, 35, 36, 39, 40, 42, 48, 50}
@Sorou...-Bastani → /workspaces/lab-4-Sorou...-Bastani (main)
○ ) $
```

Lab4f

The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder named "LAB-4-SOROUSH-BASTANI (CODE...)" containing several files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py, and README.md.
- EDITOR:** The active tab is "Lab4f.py". The code implements set operations:

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastanil
5 # Date: 2025-10-03
6 # Purpose: Practice Set Operations.
7 # Usage: ./lab4f.py
8
9 def buildtheSet(divisor):
10     result_set = set()
11
12     for num in range(0, 51):
13         if num % divisor == 0:
14             result_set.add(num)
15
16     return result_set
17
18 def s3_and_s5_not_s7(s3, s5, s7):
19     # Find numbers in BOTH s3 and s5, but NOT in s7
20     return (s3 & s5) - s7
21
22 def main():
23     s3 = buildtheSet(3)
24     print("s3: ", s3)
25     print("-----")
26
27     s5 = buildtheSet(5)
28     print("s5: ", s5)
29     print("-----")
30
31     s7 = buildtheSet(7)
32     print("s7: ", s7)
33     print("-----")
34
35     # Call the new function
36     result = s3_and_s5_not_s7(s3, s5, s7)
37     print("s3_and_s5_not_s7: ", result)
38
39 main()
```

- TERMINAL:** Shows the output of running the script:

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4f.py
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
$ ./Lab4f.py
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
s3: {0, 3, 6, 9, 12, 15, 18, 21, 24, 27, 30}
s5: {0, 35, 5, 40, 10, 45, 15, 50, 20, 25, 30}
s7: {0, 35, 7, 42, 14, 49, 21, 28}
s3_and_s5_not_s7: {45, 30, 15}
```

Lab4g

LAB-4-SOROUSH-BASTANI [CODE...]

README.md Lab4g.py

```
Lab4g.py > ...
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastanil
5 # Date: 2025-10-03
6 # Purpose: Creating dictionary.
7 # Usage: ./lab4g.py
8
9 def times_ten(start_index: int, end_index: int):
10     # Create an empty dictionary
11     my_dict = {}
12
13     # Loop through the range from start_index to end_index (inclusive)
14     for num in range(start_index, end_index + 1):
15         # Add key-value pair: key = num, value = num * 10
16         my_dict[num] = num * 10
17
18     # Return the dictionary
19     return my_dict
20
21 def main():
22     my_dictionary = times_ten(2, 6)
23     print(my_dictionary)
24
25 main()
```

CHAT

TERMINAL

Lab4h

The screenshot shows a VS Code interface with the following details:

- EXPLORER**: Shows a folder named "LAB 4-SOROUSH-BASTANI [CODE...]" containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py (selected), and README.md.
- EDITOR**: The active file is "Lab4h.py". The code contains a dictionary "my_dict" with various country and mountain pairs, and a loop that prints each key-value pair.
- TERMINAL**: Shows the command run: "/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/lab4h.py". The output lists several countries and their corresponding mountain ranges.

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python
4 # Author: Soroush Bastani sbastani
5 # Date: 2025-10-03
6 # Purpose: Traverse dictionary.
7 # Usage: ./lab4h.py
8
9 my_dict = {"switzerland" : "Alps", "United States" : "Alaska Range", "Armenia" : "Caucasus", "Argentina" : "Andes", "Pakistan" : "Kai
10
11 # Traverse the dictionary using .items()
12 for key, value in my_dict.items():
13     print(f"(key) : {value}")
```

```
$ /home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/main
$ @Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main
● ) $ /home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/main
$ /lab-4-Soroush-Bastani/ Lab4h.py
switzerland : Alps
United States : Alaska Range
Armenia : Caucasus
Argentina : Andes
Pakistan : Karakoram
$ @Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main
○ ) $
```

Lab4i

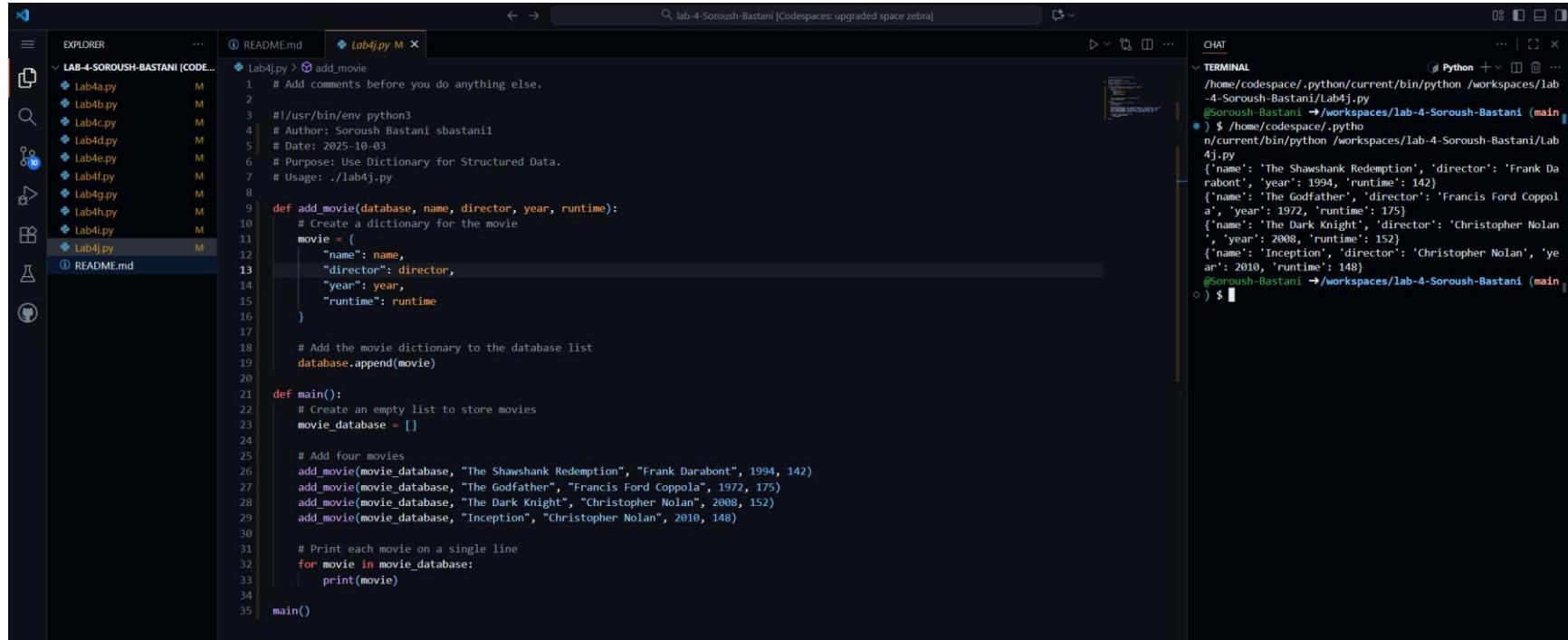
The screenshot shows a code editor interface with the following details:

- EXPLORER:** Shows a folder named "LAB-4-SOROUSH-BASTANI [CODE...]" containing files: Lab4a.py, Lab4b.py, Lab4c.py, Lab4d.py, Lab4e.py, Lab4f.py, Lab4g.py, Lab4h.py, Lab4i.py (selected), and README.md.
- Lab4i.py:** The active file in the editor. It contains Python code for creating a histogram of letters in a string. The code includes comments explaining the purpose, usage, and logic. It uses a dictionary to count letter occurrences and prints the histogram as a series of asterisks.
- TERMINAL:** Shows the command-line output of running the script. The user runs "python /lab-4-Soroush-Bastani/Lab4i.py" and the script prints a histogram for the string "hello amma". The terminal also shows the current working directory as "/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4i.py".

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani
5 # Date: 2025-10-03
6 # Purpose: Use Dictionary for Problem solving.
7 # Usage: ./Lab4i.py
8
9 def histogram(text):
10     # Create an empty dictionary to store letter counts
11     letter_count = {}
12
13     # Loop through each character in the string
14     for char in text:
15         # Skip spaces
16         if char == ' ':
17             continue
18
19         # If letter is already in dictionary, increment its count
20         if char in letter_count:
21             letter_count[char] += 1
22         # If letter is not in dictionary, add it with count 1
23         else:
24             letter_count[char] = 1
25
26         # Print the histogram
27         for letter, count in letter_count.items():
28             print(f'{letter} {"*' * count}')
29
30 def main():
31     # Call histogram with a test string
32     histogram("hello amma")
33
34 main()
```

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4i.py
@Soroush-Bastani →/workspaces/lab-4-Soroush-Bastani (main)
● ) $ ./home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4i.py
h *
e *
l **
o *
a **
m **
@Soroush-Bastani →/workspaces/lab-4-Soroush-Bastani (main)
○ ) $
```

Lab4j



The screenshot shows a terminal window with the following output:

```
/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4j.py
@Soroush-Bastani → /workspaces/lab-4-Soroush-Bastani (main)
$ /home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4j.py
[{'name': 'The Shawshank Redemption', 'director': 'Frank Darabont', 'year': 1994, 'runtime': 142}, {'name': 'The Godfather', 'director': 'Francis Ford Coppola', 'year': 1972, 'runtime': 175}, {'name': 'The Dark Knight', 'director': 'Christopher Nolan', 'year': 2008, 'runtime': 152}, {'name': 'Inception', 'director': 'Christopher Nolan', 'year': 2010, 'runtime': 148}]
```

The terminal window is titled "TERMINAL" and shows the command `/home/codespace/.python/current/bin/python /workspaces/lab-4-Soroush-Bastani/Lab4j.py` being run. The output displays a list of four movie dictionaries.

The code in the editor is as follows:

```
1 # Add comments before you do anything else.
2
3 #!/usr/bin/env python3
4 # Author: Soroush Bastani sbastani1
5 # Date: 2025-10-03
6 # Purpose: Use Dictionary for Structured Data.
7 # Usage: ./lab4j.py
8
9 def add_movie(database, name, director, year, runtime):
10     # Create a dictionary for the movie
11     movie = {
12         "name": name,
13         "director": director,
14         "year": year,
15         "runtime": runtime
16     }
17
18     # Add the movie dictionary to the database list
19     database.append(movie)
20
21 def main():
22     # Create an empty list to store movies
23     movie_database = []
24
25     # Add four movies
26     add_movie(movie_database, "The Shawshank Redemption", "Frank Darabont", 1994, 142)
27     add_movie(movie_database, "The Godfather", "Francis Ford Coppola", 1972, 175)
28     add_movie(movie_database, "The Dark Knight", "Christopher Nolan", 2008, 152)
29     add_movie(movie_database, "Inception", "Christopher Nolan", 2010, 148)
30
31     # Print each movie on a single line
32     for movie in movie_database:
33         print(movie)
34
35 main()
```