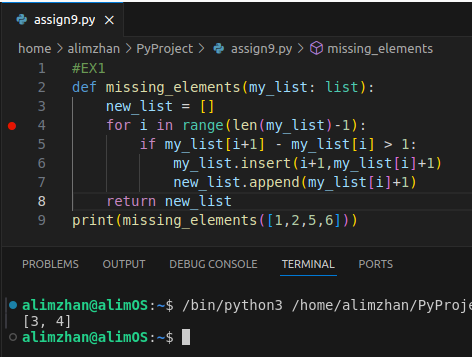
Exercise-1: Find missing elements

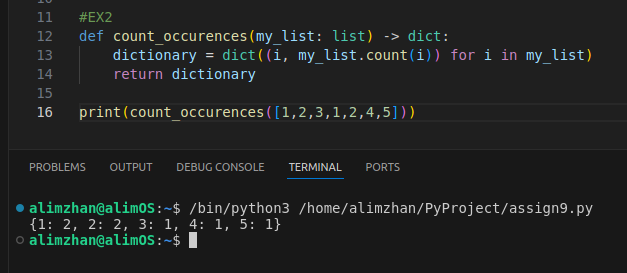
Write a function "missing\_elements(my\_list: list) -> list" that takes a

sorted list of integers and returns a list of missing integers in the range of the list.



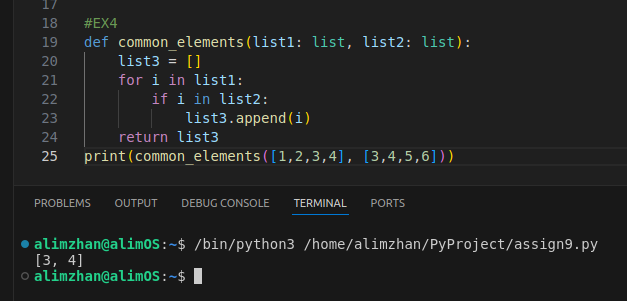
Exercise-2: Count occurrences

Write a function "count\_occurrences(my\_list: list) -> dict" that takes a list of integers and returns a dictionary where keys are unique integers from the list and values are their counts in the list.



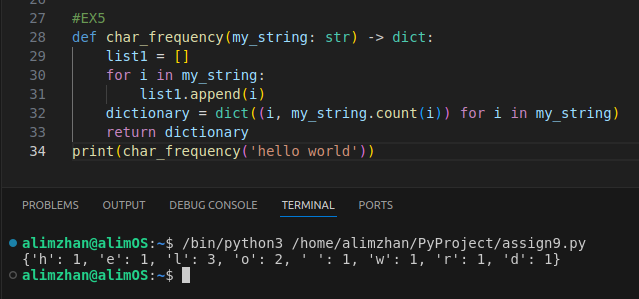
Exercise-4: Common elements

Write a function "common\_elements(list1: list, list2: list) -> list" that takes two lists of integers and returns a list of unique common elements.



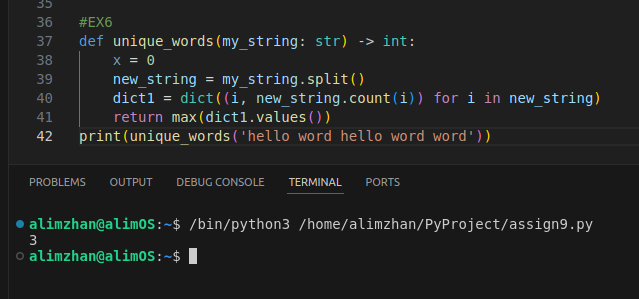
Exercise-5: Character frequency

Write a function "char\_frequency(my\_string: str) -> dict" that takes a string and returns a dictionary with the frequency of each character in the string.



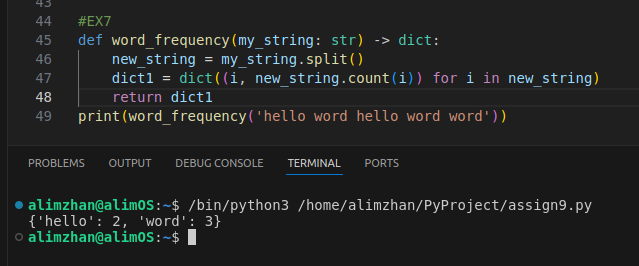
Exercise-6: Unique words

Write a function "unique\_words(my\_string: str) -> int" that takes a string and returns the number of unique words in the string.



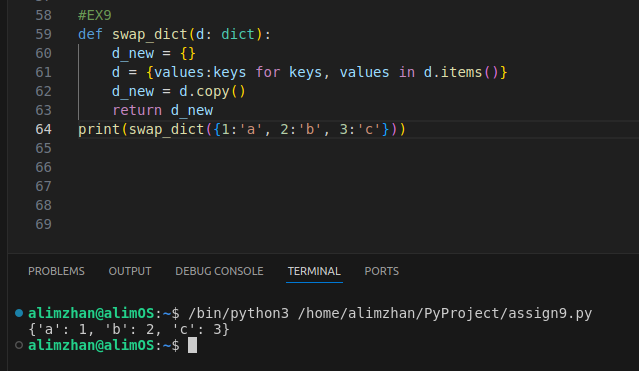
Exercise-7: Word frequency

Write a function "word\_frequency(my\_string: str) -> dict" that takes a string and returns a dictionary with the frequency of each word in the string.



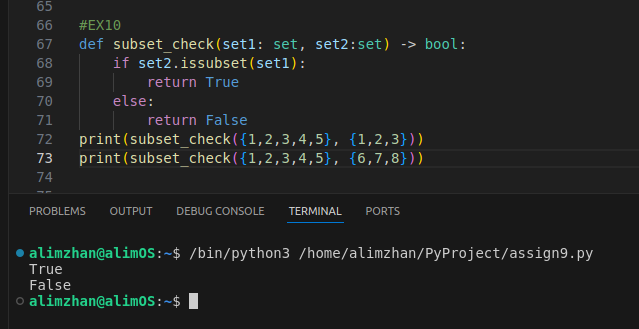
Exercise-9: Swap dictionary keys and values

Write a function "swap\_dict(d: dict) -> dict" that takes a dictionary and returns a new dictionary where keys become values and values become keys. if you face duplicates, the first key should be saved.



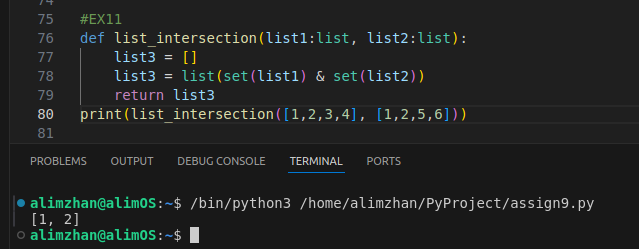
Exercise-10: Subset check

Write a function "is\_subset(set1: set, set2: set) -> bool" that takes two sets and returns True if set2 is a subset of set1, and False otherwise.



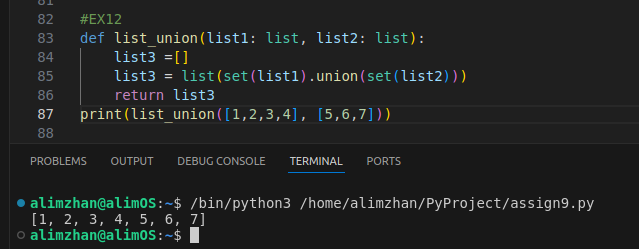
Exercise-11: Intersection of lists

Write a function "list\_intersection(list1: list, list2: list) -> list" that takes two lists and returns a list of unique elements that are in both lists.



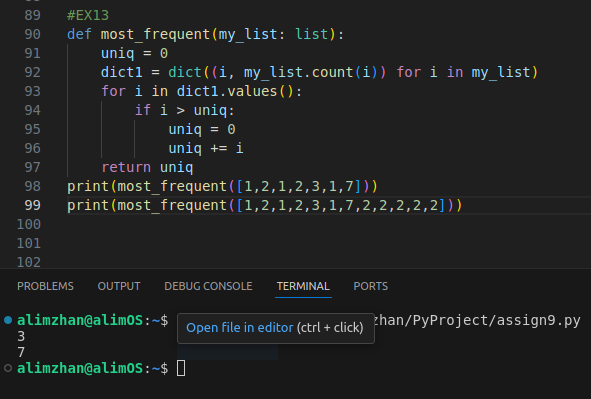
Exercise-12: Union of lists

Write a function "list\_union(list1: list, list2: list) -> list" that takes two lists and returns a list of unique elements that are in either of the lists.



Exercise-13: Most frequent element

Write a function "most\_frequent(my\_list: list) -> int" that takes a list of integers and returns the most frequent element in the list.



Exercise-14: Least frequent element

Write a function "least\_frequent(my\_list: list) -> int" that takes a list of integers and returns the least frequent element in the list.

