```
#1:
def unique_sorted_letters(input_string):
    # Convert the string to lowercase to ensure uniqueness is case-insensitive
    input_string = input_string.lower()
      # Use a set to get unique letters
    unique_letters = set(input_string)
    # Filter out non-letter characters and sort the result
    sorted_unique_letters = sorted([char for char in unique_letters if
char.isalpha()])

    return sorted_unique_letters

print(unique_sorted_letters("uniqueness of a string"))
print(unique_sorted_letters("984157"))
```

['a', 'e', 'f', 'g', 'i', 'n', 'o', 'q', 'r', 's', 't', 'u']

```
2:
def letters_in_either(word1, word2):
    """Return a sorted list of letters that appear in at least one of the two
words."""
    return sorted(set(word1) | set(word2))

def letters_in_both(word1, word2):
    """Return a sorted list of letters that appear in both words."""
    return sorted(set(word1) & set(word2))

def letters_in_either_not_both(word1, word2):
    """Return a sorted list of letters that appear in either word, but not in
both."""
    return sorted(set(word1) ^ set(word2))

print(letters_in_either("application", "aliases"))
print(letters_in_both("portability", "picsart"))
print(letters_in_either_not_both("hello", "worldd"))
```

['a', 'c', 'e', 'i', 'l', 'n', 'o', 'p', 's', 't']

['a', 'i', 'p', 'r', 't']

['d', 'e', 'h', 'r', 'w']

```python
#3:
def main():
    # Dictionary to store countries and their capitals
    countries_capitals = {}

    while True:
        country = input("Enter the name of a country (or type 'exit' to quit):
").strip() #strip() method is used to remove any leading or trailing whitespace
from the input.

        # Check if the user wants to exit
        if country.lower() == 'exit':
            print("Exiting the program")
            break

        # Check if the country is already in the dictionary
        if country in countries_capitals:
            print(f"The capital of {country} is {countries_capitals[country]}.")
        else:
            # If the capital is not known, ask the user to enter it
            capital = input(f"Enter the capital of {country}: ").strip()
            countries_capitals[country] = capital  # adds new entry to the
countries_capitals dictionary
            print(f"Thank you!  {capital} has been added as the capital of
{country}.")
main()
```

Enter the name of a country (or type 'exit' to quit): nepal

Enter the capital of nepal: kathmandu

Thank you! kathmandu has been added as the capital of nepal.

Enter the name of a country (or type 'exit' to quit): nepal

The capital of nepal is kathmandu.

Enter the name of a country (or type 'exit' to quit): germany

Enter the capital of germany: berlin

Thank you! berlin has been added as the capital of germany.

Enter the name of a country (or type 'exit' to quit):

```
4:
def frequency_analysis(message):
    # Normalize the message to lowercase
    message = message.lower()

    # Dictionary to store letter counts
    letter_count = {}

    # Count occurrences of each letter
    for char in message:
        if char.isalpha():  # Check if the character is a letter
            if char in letter_count:
                letter_count[char] += 1
            else:
                letter_count[char] = 1

    # Sort the letters by frequency (highest first) and then alphabetically
    sorted_letters = sorted(letter_count.items(), key=lambda item: (-item[1],
item[0]))#tuple for sorting

    # Get the top six most common letters
    top_six = sorted_letters[:6]

    # Print the results
    print("The six most common letters are:")
    for letter, count in top_six:
        print(f"'{letter}': {count}")

# Example usage
message = "This is an example message to demonstrate frequency analysis."
frequency_analysis(message)
```

The six most common letters are:

'e': 8

's': 7

'a': 6

'n': 4

't': 4

'i': 3