```
# Write and test a function that takes a string as a parameter and
returns a sorted list of all the unique letters used in the string.
So, if th
# string is cheese, the list returned should be ['c', 'e', 'h', 's']

def unique_letters(string):
    return sorted(set(string))

input_string= "cheese"
output= unique_letters(input_string)
print(f"The sorted list is {output}")
```

```
The sorted list is ['c', 'e', 'h', 's']
```

```
#  Write and test three functions that each take two words (strings)
as parameters and return sorted lists (as defined above) representing
respectively:
# Letters that appear in at least one of the two words.
# Letters that appear in both words.
# Letters that appear in either word, but not in both.


def sort_letters(para1,para2):
    return sorted((set(para1+para2)))

def both_words(para1,para2):
    return sorted(set(para1) & set(para2))

def either_or_both(para1,para2):
     return sorted(set(para1) & set(para2))

para1="greet"
para2="read"

print(f"Letter that appear atlest one of the two
words{sort_letters(para1,para2)}")
print(f"letter that appear in both words{both_words(para1,para2)}")
print(f"Letter that appear in either word,but not in
both{para1,para2}")
```

```
Letter that appear atlest one of the two words['a', 'd', 'e', 'g',
'r', 't']
letter that appear in both words['e', 'r']
Letter that appear in either word,but not in both('greet', 'read')
```

```
# Write a program that manages a list of countries and their capital
cities. It should prompt the user to enter the name of a country. If
the program
# already "knows" the name of the capital city, it should display it.
Otherwise it should ask the user to enter it.
```

```
countries_with_capitals={'Nepal':'Kathmandu','Korea':'Seoul','China':'
Beijing','Japan':'Tokyo','France':'Paris'}

country=input("Enter the country you want to know the capital about :
")
if country in countries_with_capitals:
    print(f"The capital of {country} is:
{countries_with_capitals[country]}")
else:
    print("Sorry, the capital for the country is not stored.")
```

```
Enter the country you want to know the capital about :  Korea

The capital of Korea is: Seoul
```

```
# One approach to analysing some encrypted data where a substitution
is suspected is frequency analysis. A count of the different symbols
in the message
# can be us to identify the language used, and sometimes some of the
letters. In English, the most common letter is "e", and so the symbol
representing "
# e" should appear most in the encrypted text.

def letter_counting(message):
    letter_count = {}

    for char in message.lower():
        if 'a' <= char <= 'z':
            letter_count[char] = letter_count.get(char, 0) + 1

    sorted_letters = sorted(letter_count.items(), key=lambda item:
item[1], reverse=True)

    for letter, count in sorted_letters[:6]:
        print(f"{letter}: {count} times")


message = "Wishing you a prosperous life !"

letter_counting(message)
```

```
i: 3 times
s: 3 times
o: 3 times
u: 2 times
p: 2 times
r: 2 times
```