

```
letters = "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
```

```
# Write your unique_english_letters function here:
```

```
def unique_english_letters(word):
```

```
    uniques = 0
```

```
    for letter in letters:
```

```
        if letter in word:
```

```
            uniques += 1
```



```
    return uniques
```

```
# Uncomment these function calls to test your function:
```

```
print(unique_english_letters("mississippi"))
```



```
# should print 4
```

```
print(unique_english_letters("Apple"))
```


 My Home Course Menu Get Unstuck 

▶ Hint

```
1 letters =
  "ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz"
2 # Write your unique_english_letters function
  here:
3 def unique_english_letters(word):
4     uniques = 0
5     for letter in letters:
6         if letter in word:
7             uniques += 1
8     return uniques
9 # Uncomment these function calls to test your
  function:
10 print(unique_english_letters("mississippi"))
11 # should print 4
12 print(unique_english_letters("Apple"))
13 # should print 4
```

Run  

4
4

 You got it!

Back Next

```
# should print 4
```

2.

```
# Write your count_char_x function here:
```

```
def count_char_x(word, x):
```

```
    occurrences = 0
```

```
    for letter in word:
```



```
        if letter == x:
```

```
            occurrences += 1
```

```
    return occurrences
```

```
# Uncomment these function calls to test your tip function:
```

```
print(count_char_x("mississippi", "s"))
# should print 4
print(count_char_x("mississippi", "m"))
# should print 1
```

 My Home Course Menu Get Unstuck 


► Hint


```
1 # Write your count_char_x function here:
2 ▼ def count_char_x(word, x):
3     occurrences = 0
4     ▼ for letter in word:
5     ▼     if letter == x:
6         occurrences += 1
7     return occurrences
8 # Uncomment these function calls to test your
  tip function:
9 print(count_char_x("mississippi", "s"))
10 # should print 4
11 print(count_char_x("mississippi", "n"))
12 # should print 1
```


4

1

Run






 You got it!


Back

Next

3. # Write your count_multi_char_x function here:

```
def count_multi_char_x(word, x):
    splits = word.split(x)
    return(len(splits)-1)
# Uncomment these function calls to test your function:
print(count_multi_char_x("mississippi", "iss"))
# should print 2
print(count_multi_char_x("apple", "pp"))
# should print 1
```



 My Home Course Menu


Get Unstuck

▶ Hint

```
1 # Write your count_multi_char_x function here:
2 def count_multi_char_x(word, x):
3     splits = word.split(x)
4     return len(splits)-1
5 # Uncomment these function calls to test your
  function:
6 print(count_multi_char_x("mississippi", "iss")
7 )
8 # should print 2
9 print(count_multi_char_x("apple", "pp"))
10 # should print 1
```

Run




 You got it!

BackNext

4. # Write your substring_between_letters function here:



```
def substring_between_letters(word, start, end):
    start_ind = word.find(start)
    end_ind = word.find(end)
    if start_ind > -1 and end_ind > -1:
        return word[start_ind+1:end_ind]
    return word
# Uncomment these function calls to test your function:
print(substring_between_letters("apple", "p", "e"))
# should print "pl"
print(substring_between_letters("apple", "p", "c"))
# should print "apple"
```

My HomeCourse Menu


Get Unstuck

Time

```
1 # Write your substring_between_letters
  function here:
2 def substring_between_letters(word, start,
  end):
3     start_ind = word.find(start)
4     end_ind = word.find(end)
5     if start_ind > -1 and end_ind > -1:
6         return word[start_ind+1:end_ind]
7     return word
8 # Uncomment these function calls to test your
  function:
9 print(substring_between_letters("apple", "p",
  "e"))
10 # should print "pl"
11 print(substring_between_letters("apple", "p",
  "c"))
12 # should print "apple"
```

Run

pl
apple

 You got it!

Back

Next

5. `# Write your x_length_words function here:`

```
def x_length_words(sentence, x):
```

```
    words = sentence.split(" ")
```

```
    for word in words:
```

```
        if len(word) < x:
```

```
            return False
```

```
    return True
```

```
# Uncomment these function calls to test your tip function:
```

```
print(x_length_words("i like apples", 2))
```

```
# should print False
```

```
print(x_length_words("he likes apples", 2))
```

```
# should print True
```



`sentence` has a length greater than or equal to `x`.

► Hint

```
1 # Write your x_length_words function here:
2 def x_length_words(sentence, x):
3     words = sentence.split(" ")
4     for word in words:
5         if len(word) < x:
6             return False
7     return True
8 # Uncomment these function calls to test your
  tip function:
9 print(x_length_words("i like apples", 2))
10 # should print False
11 print(x_length_words("he likes apples", 2))
12 # should print True
```

False
True

Run



You got it!

Back

Next