

Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>>

= RESTART: C:\Users\Tamale Simon Peter\Documents\--UoPeople\CS1101\unit  
7\Programming\_assignment.py

alphabet = "abcdefghijklmnopqrstuvwxyz"

test\_dups = ["zzz", "dog", "bookkeeper", "subdermatoglyphic", "subdermatoglyphics"]

test\_miss = ["zzz", "subdermatoglyphic", "the quick brown fox jumps over the lazy dog"]

def histogram(s):

d = dict()

for c in s:

if c not in d:

d[c] = 1

else:

d[c] += 1

return d

""" Part 1 """

def has\_duplicates(s): #has\_duplicates function definition

my\_dict = histogram(s)

for v in my\_dict.values():

if v > 1:

return True

return False

```

def test_dups_func(): #testing the duplicates
    for s in test_dups:
        duplicates = has_duplicates(s)
        if duplicates:
            print (s+ ' ', 'has duplicates', '\n')
        else:
            print (s+ ' ', 'has no duplicates', '\n')

```

```

def missing_letters(s):
    r = list(alphabet)
    s = s.lower()
    for c in s.lower():
        if c in r:
            r.remove(c) # removes the instance where there is no missing_letters
    return ".join(r)

```

""" Part 2 """

```

def test_miss_func():

    for s in test_miss:
        missing = missing_letters(s)
        if missing:
            print(s + ' ' + 'is missing leters', missing_letters(s), '\n')
        else:
            print(s+ ' ' + 'uses all the letters',)

```

Output

```
>>> test_dups_func()
```

```
zzz has duplicates
```

```
dog has no duplicates
```

```
bookkeeper has no duplicates
```

```
subdermatoglyphic has no duplicates
```

```
subdermatoglyphics has duplicates
```

```
>>> test_miss_func()
```

```
zzz is missing letters abcdefghijklmnopqrstuvwxyz
```

```
subdermatoglyphic is missing letters fjkqvwxyz
```

```
the quick brown fox jumps over the lazy dog uses all the letters
```

```
>>>
```