

```
"""
Rashad Khan
010713326
CS2520-01
Lab 7
03/27/2023
"""
```

```
from collections import Counter
#task 1
```

```
test_input =((input("Please entere a pargraph of text: \n")).lower()).split()
test_dict = {}
```

```
for i in test_input:
    if i in test_dict:
        test_dict[i] += 1
    else:
        test_dict.update({i:1})
print ("\n",test_dict)
highest = Counter(test_dict)
high3 = highest.most_common(3)
```

```
print("\n",high3)
```

```
"""
Please entere a pargraph of text:
python is an easy language python is easy to learn and easy to code a lot of python modules that are
easy to use go python
```

```
{'python': 4, 'is': 2, 'an': 1, 'easy': 4, 'language': 1, 'to': 3, 'learn': 1, 'and': 1, 'code': 1,
'a': 1, 'lot': 1, 'of': 1, 'modules': 1, 'that': 1, 'are': 1, 'use': 1, 'go': 1}
```

```
[('python', 4), ('easy', 4), ('to', 3)]
```

```
"""
Please entere a pargraph of text:
When choosing a collection type, it is useful to understand the properties of that type. Choosing the
right type for a particular data set could mean retention of meaning, and, it could mean an increase
in efficiency or security.
```

```
{'when': 1, 'choosing': 2, 'a': 2, 'collection': 1, 'type': 1, 'it': 2, 'is': 1, 'useful': 1, 'to':
1, 'understand': 1, 'the': 2, 'properties': 1, 'of': 2, 'that': 1, 'type.': 1, 'right': 1, 'type': 1,
'for': 1, 'particular': 1, 'data': 1, 'set': 1, 'could': 2, 'mean': 2, 'retention': 1, 'meaning.': 1,
'and.': 1, 'an': 1, 'increase': 1, 'in': 1, 'efficiency': 1, 'or': 1, 'security.': 1}
```

```
[('choosing', 2), ('a', 2), ('it', 2)]
```

```
"""
Please entere a pargraph of text:
to test or not to test that is the true question for ages testers have thought if it is worth testing
their code before turinging anything ind but sadly after not testing many people have started failing
assignments and realising that not testing is the
downfall of the peoples grades
```

```
{'to': 2, 'test': 2, 'or': 1, 'not': 3, 'that': 2, 'is': 3, 'the': 3, 'true': 1, 'question': 1,
'for': 1, 'ages': 1, 'testers': 1, 'have': 2, 'thought': 1, 'if': 1, 'it': 1, 'worth': 1, 'testing':
3, 'their': 1, 'code': 1, 'before': 1, 'turinging': 1, 'anything': 1, 'ind': 1, 'but': 1, 'sadly': 1,
'after': 1, 'many': 1, 'people': 1, 'started': 1, 'failing': 1, 'assignments': 1, 'and': 1,
```

```

'realising': 1, 'downfall': 1, 'of': 1, 'peoples': 1, 'grades': 1}

[('not', 3), ('is', 3), ('the', 3)]
"""

import random
#task 2

L1 = [random.randrange(1,101) for i in range(1, 101)]
L2 = [i for i in range(1, 101) if i % 3 == 0 or i % 4 == 0]

#print(L1)
#print(L2)

S1= set(L1)
S2= frozenset(L2)

#print(S1)
#print(S2)

R1 = S1.union(S2)

print(len(R1))

R2 = S1.intersection(S2)

print(len(R2))

R3 = S1.difference(S2)

print(len(R3))

"""
81
33
31
"""

```