1.In this example, words of the same length are grouped together in lists, and the keys in the dictionary represent the word lengths.

# Sample list

words = ['apple', 'banana', 'cherry', 'date', 'fig', 'grape']

# Create an empty dictionary to store words by their lengths

length\_dict = {}

# Iterate through the list of words

for word in words:

length = len(word)

if length not in length\_dict:

length\_dict[length] = []

length\_dict[length].append(word)

print(length\_dict)

output:

{5: ['apple', 'grape'], 6: ['banana', 'cherry'], 4: ['date'], 3: ['fig']}

2.Dictonary to List with tuples

d = {'a': 1, 'b': 2, 'c': 3}

# Convert to list of tuples

data = list(d.items())

print(data)

3. wordcount program

f = open("wordcount.txt", "r")

paragraph = f.read()

words = paragraph.split()

print(words)

print(type(words))

p=[]

for x in words:

p.append(x)

print(p)

# Create a dictionary to store word lengths and their counts

wlength = {}

for x in words:

length = len(x)

#print(length)

if length in wlength:

wlength[length] += 1

else:

wlength[length] = 1

print(wlength)

def segregate\_words\_by\_length(words):

word\_dict = {}

for word in words:

length = len(word)

if length not in word\_dict:

word\_dict[length] = []

word\_dict[length].append(word)

return word\_dict

segregated\_words = segregate\_words\_by\_length(words)

print(segregated\_words)