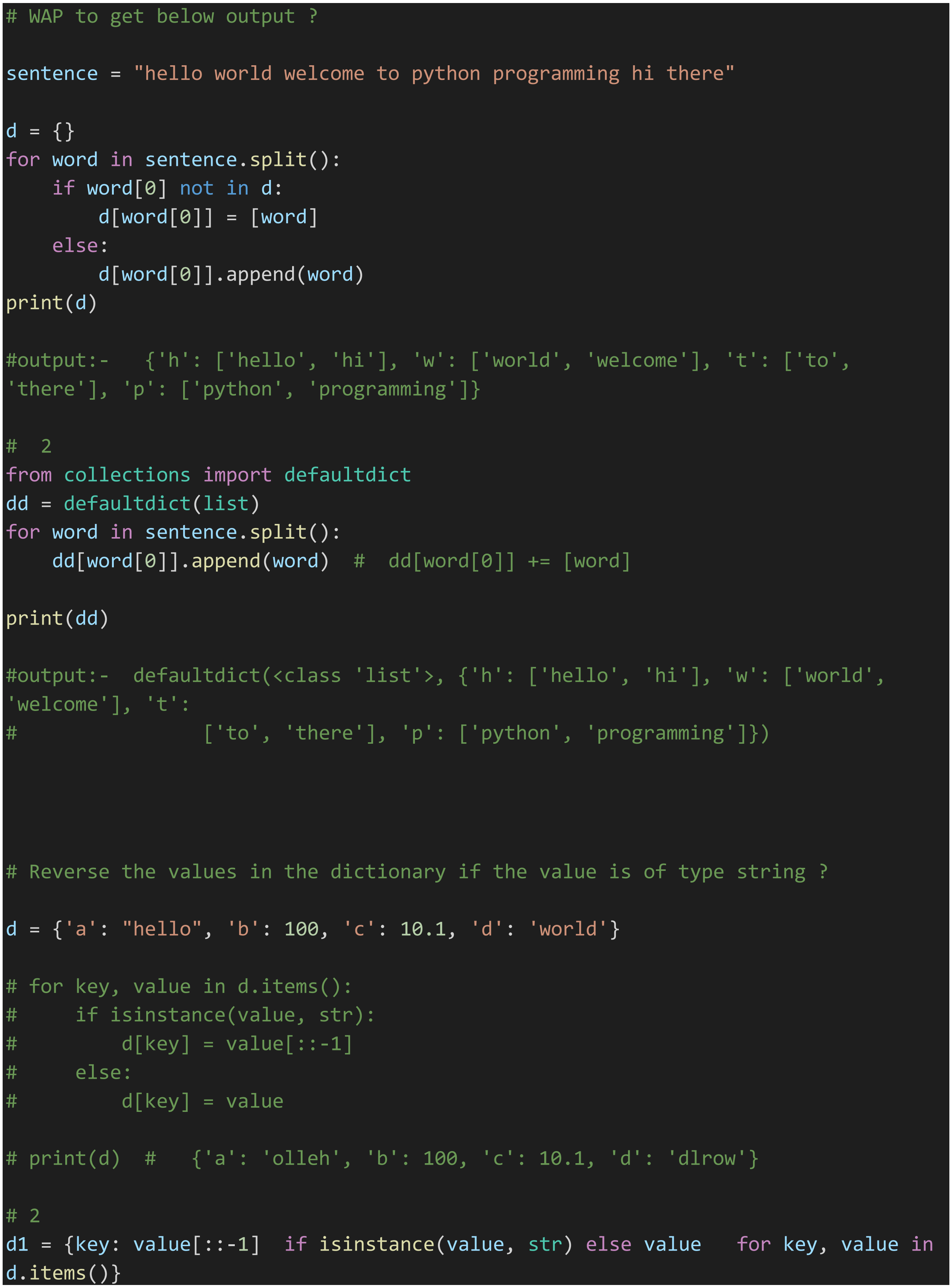
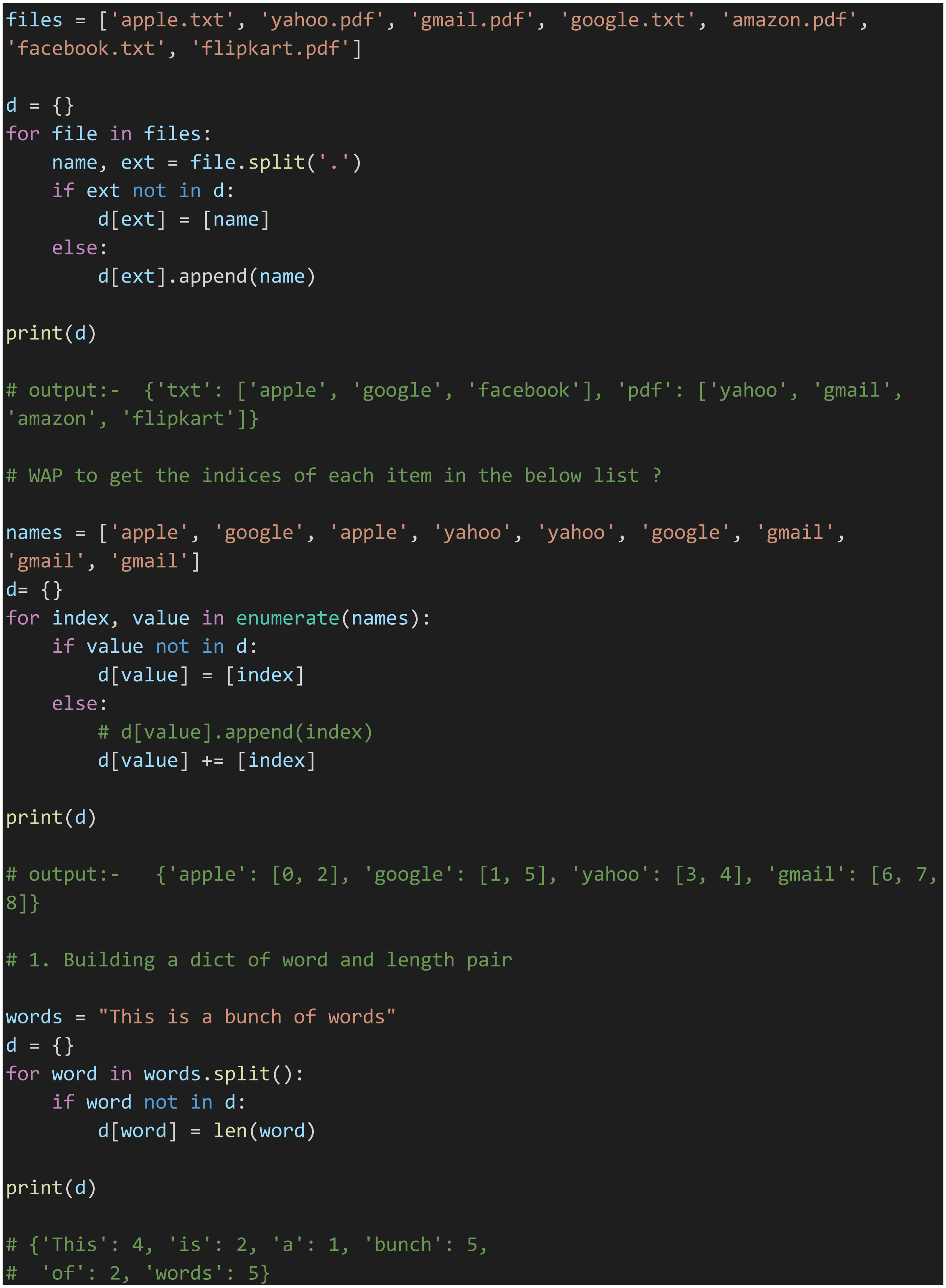
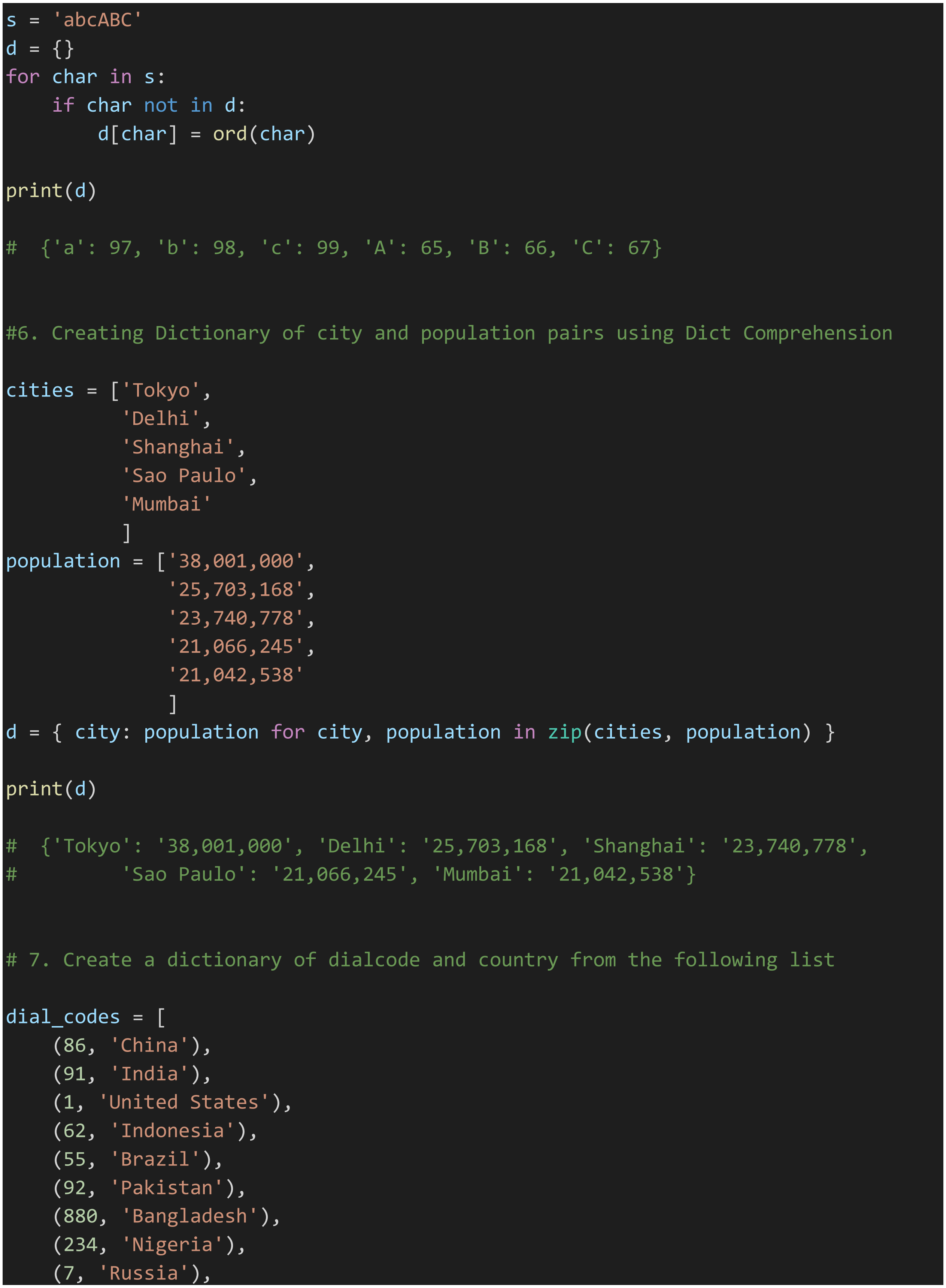


```
# WAP to get below output ?

sentence = "hello world welcome to python programming hi there"

d = {}
for word in sentence.split():
    if word[0] not in d:
        d[word[0]] = [word]
    else:
        d[word[0]].append(word)
print(d)

#output:-   {'h': ['hello', 'hi'], 'w': ['world', 'welcome'], 't': ['to',
'there'], 'p': ['python', 'programming']}

#  2
from collections import defaultdict
dd = defaultdict(list)
for word in sentence.split():
    dd[word[0]].append(word)  #  dd[word[0]] += [word]

print(dd)

#output:-  defaultdict(<class 'list'>, {'h': ['hello', 'hi'], 'w': ['world',
'welcome'], 't':
#                ['to', 'there'], 'p': ['python', 'programming']})




# Reverse the values in the dictionary if the value is of type string ?

d = {'a': "hello", 'b': 100, 'c': 10.1, 'd': 'world'}

# for key, value in d.items():
#     if isinstance(value, str):
#         d[key] = value[::-1]
#     else:
#         d[key] = value

# print(d)  #   {'a': 'olleh', 'b': 100, 'c': 10.1, 'd': 'dlrow'}

# 2
d1 = {key: value[::-1]  if isinstance(value, str) else value   for key, value in
d.items()}
```

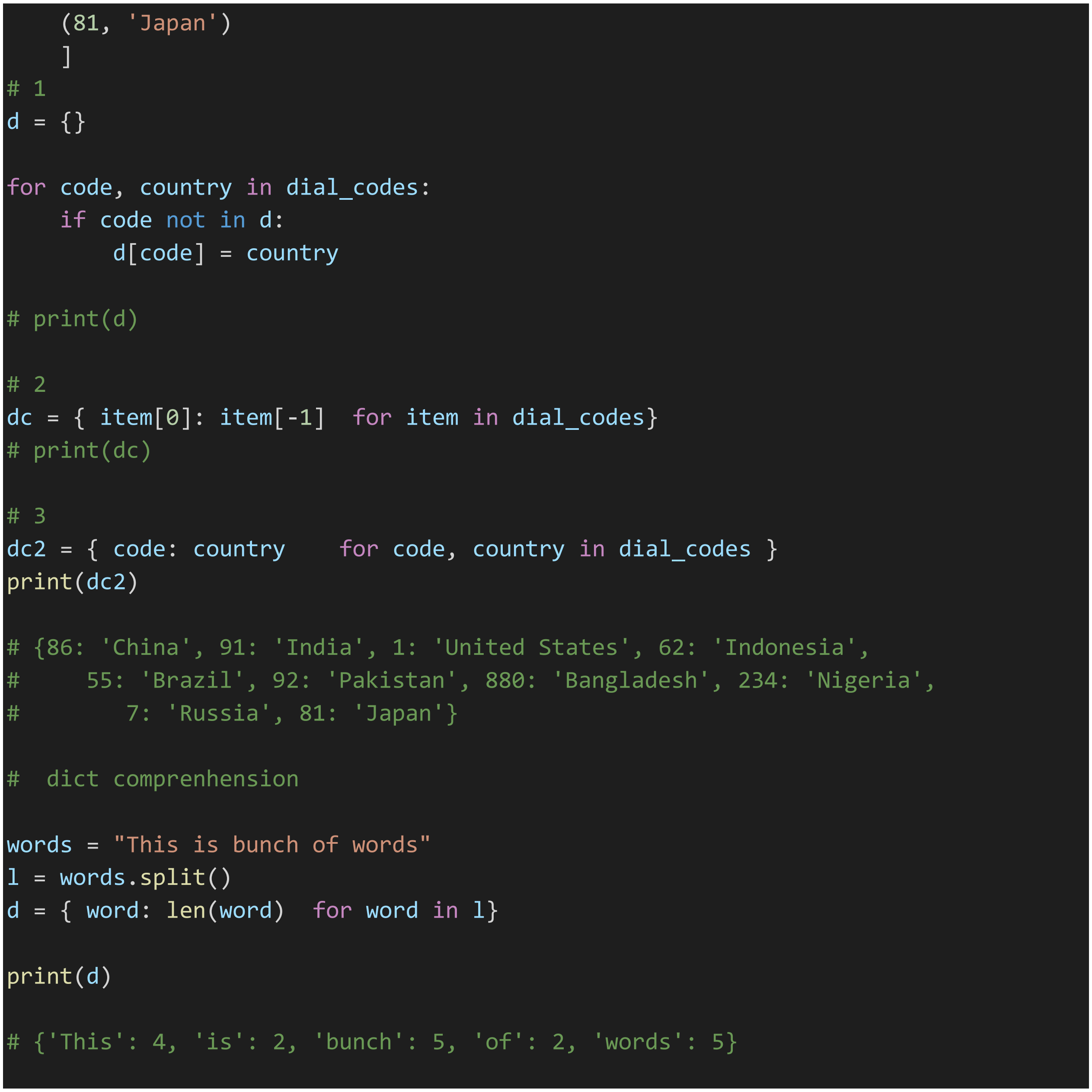
```
print(d1)

# output:-

# WAP to replace value present in nested dictionary. Replace "nose" with "net" ?

d= {'a': 100, 'b': {'m': 'man', 'n': 'nose', 'o': 'ox', 'c': 'cat'}}

def replace_dict(dict_, old_, new_):
    for key, value in dict_.items():
        if isinstance(value, dict):
            for k, v in value.items():
                if v == old_:
                    value[k] = new_

    return dict_

res = replace_dict(d, 'nose', 'net')
print(res)

# output:-  {'a': 100, 'b': {'m': 'man', 'n': 'net', 'o': 'ox', 'c': 'cat'}}

# Group flower and Animals in the below list ?

items = ['lotus-flower', 'lilly-flower', 'cat-animal','sunflower-flower', 'dog-
animal']

d= {}
for item in items:
    name, grp = item.split("-")
    if grp not in d:
        d[grp] = [name]
    else:
        # d[grp].append(name)
         d[grp] += [name]

print(d)

# output:-   {'flower': ['lotus', 'lilly', 'sunflower'], 'animal': ['cat',
'dog']}


#  Grouping files with same extension
```



```
files = ['apple.txt', 'yahoo.pdf', 'gmail.pdf', 'google.txt', 'amazon.pdf',
'facebook.txt', 'flipkart.pdf']

d = {}
for file in files:
    name, ext = file.split('.')
    if ext not in d:
        d[ext] = [name]
    else:
        d[ext].append(name)

print(d)

# output:-  {'txt': ['apple', 'google', 'facebook'], 'pdf': ['yahoo', 'gmail',
'amazon', 'flipkart']}

# WAP to get the indices of each item in the below list ?

names = ['apple', 'google', 'apple', 'yahoo', 'yahoo', 'google', 'gmail',
'gmail', 'gmail']
d= {}
for index, value in enumerate(names):
    if value not in d:
        d[value] = [index]
    else:
        # d[value].append(index)
        d[value] += [index]

print(d)

# output:-   {'apple': [0, 2], 'google': [1, 5], 'yahoo': [3, 4], 'gmail': [6, 7,
8]}

# 1. Building a dict of word and length pair

words = "This is a bunch of words"
d = {}
for word in words.split():
    if word not in d:
        d[word] = len(word)

print(d)

# {'This': 4, 'is': 2, 'a': 1, 'bunch': 5,
#  'of': 2, 'words': 5}
# 2. Flipping keys and values of the dictionary using dict comprehension

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

flip_dict = {value: key  for key, value in d.items()  }

print(flip_dict)

# output:- {1: 'a', 2: 'b', 3: 'c'}

# 3. Counting the number of each character in a String

my_string = 'guido van rossum'

d = {}
for char in my_string:
    if char not in d:
        d[char] = 1
    else:
        d[char] += 1

print(d)

# output:-  {'g': 1, 'u': 2, 'i': 1, 'd': 1, 'o': 2, ' ': 2, 'v': 1, 'a': 1, 'n':
1, 'r': 1, 's': 2, 'm': 1}

# 4. Counting the number of each character in a String

sentence = "hello world welcome to python hello hi world welcome to python"

d = {}
for char in sentence:
    if char not in d:
        d[char] = 1
    else:
        d[char] += 1

print(d)

# output:- {'h': 5, 'e': 6, 'l': 8, 'o': 10, ' ': 10, 'w': 4, 'r': 2, 'd': 2,
'c': 2, 'm': 2, 't': 4, 'p': 2, 'y': 2, 'n': 2, 'i': 1}

# 5. Dictionary of character and ascii value pairs
```



```python
s = 'abcABC'
d = {}
for char in s:
    if char not in d:
        d[char] = ord(char)

print(d)

#  {'a': 97, 'b': 98, 'c': 99, 'A': 65, 'B': 66, 'C': 67}


#6. Creating Dictionary of city and population pairs using Dict Comprehension

cities = ['Tokyo',
          'Delhi',
          'Shanghai',
          'Sao Paulo',
          'Mumbai'
          ]
population = ['38,001,000',
              '25,703,168',
              '23,740,778',
              '21,066,245',
              '21,042,538'
              ]
d = { city: population for city, population in zip(cities, population) }

print(d)

#  {'Tokyo': '38,001,000', 'Delhi': '25,703,168', 'Shanghai': '23,740,778',
#         'Sao Paulo': '21,066,245', 'Mumbai': '21,042,538'}


# 7. Create a dictionary of dialcode and country from the following list

dial_codes = [
    (86, 'China'),
    (91, 'India'),
    (1, 'United States'),
    (62, 'Indonesia'),
    (55, 'Brazil'),
    (92, 'Pakistan'),
    (880, 'Bangladesh'),
    (234, 'Nigeria'),
    (7, 'Russia'),
```



```
    (81, 'Japan')
    ]
# 1
d = {}

for code, country in dial_codes:
    if code not in d:
        d[code] = country

# print(d)

# 2
dc = { item[0]: item[-1]  for item in dial_codes}
# print(dc)

# 3
dc2 = { code: country    for code, country in dial_codes }
print(dc2)

# {86: 'China', 91: 'India', 1: 'United States', 62: 'Indonesia',
#     55: 'Brazil', 92: 'Pakistan', 880: 'Bangladesh', 234: 'Nigeria',
#        7: 'Russia', 81: 'Japan'}

#  dict comprenhension

words = "This is bunch of words"
l = words.split()
d = { word: len(word)  for word in l}

print(d)

# {'This': 4, 'is': 2, 'bunch': 5, 'of': 2, 'words': 5}
```