```python
#!/usr/bin/env python
# coding: utf-8

# In[ ]:


message = 'Call me at 415-555-1011 tomorrow. 415-555-9999 is my office.'

# Returns True if input in a phone number, else returns False.
def isPhoneNumber(text):
    if len(text) != 12:
        return False
    for i in range(0, 3):
        if not text[i].isdecimal():
            return False
    if text[3] != '-':
        return False
    for i in range(4, 7):
        if not text[i].isdecimal():
            return False
    if text[7] != '-':
        return False
    for i in range(8, 12):
        if not text[i].isdecimal():
            return False
    return True

# Searches through a string for phone numbers.
def num_search(string):
    for i in range(len(string)):
        chunk = string[i:i+12]
        if isPhoneNumber(chunk):
            print('Phone number found: ' + chunk)
    print('Done')

print(message + '\n')
num_search(message)


# ### re.compile(), .search(), .group(), .groups()

# In[ ]:


import re
```

```
46
47  message = 'My number is 415-555-4242.'
48  message_1 = 'My phone number is (415)-555-4242.'
49
50  # Finds phone number in message
51
52  print('Regex: (\d\d\d)-(\d\d\d-\d\d\d\d)')
53  regex = re.compile(r'(\d\d\d)-(\d\d\d-\d\d\d\d)')
54  mo = regex.search(message)
55  print(mo.group() + '\n')
56
57  print('Regex: (\(\d\d\d\))-(\d\d\d-\d\d\d\d)')
58  # Finds phone number in message_1.
59  regex = re.compile(r'(\(\d\d\d\))-(\d\d\d-\d\d\d\d)')
60  mo = regex.search(message_1)
61  print(mo.group())
62
63
64  # ### | (pipe)
65
66  # In[ ]:
67
68
69  message = 'My number is 415-555-4242.'
70  message_1 = 'My phone number is 415 555 4242.'
71
72  # Finds phone number in message and message_1. Uses pipe to indicate a - or ' '.
73  print('Regex: ((\d\d\d)(-|\s)\d\d\d(-|\s)\d\d\d\d)')
74  for item in [message, message_1]:
75      regex = re.compile(r'((\d\d\d)(-|\s)\d\d\d(-|\s)\d\d\d\d)')
76      mo = regex.search(item)
77      print(mo.group())
78
79
80  # ### ? (optional)
81
82  # In[ ]:
83
84
85  message = 'My number is 415-555-4242.'
86  message_1 = 'My phone number is 415 555 4242.'
87  message_2 = 'My phone number is 4155554242.'
88
89  # Finds phone number in message, message_1, and message_2. Uses ? to indicate - or ' ' is optional.
90  print('Regex: ((\d\d\d)(-|\s)?\d\d\d(-|\s)?\d\d\d\d)')
```

```
for item in [message, message_1, message_2]:
    regex = re.compile(r'((\d\d\d)(-|\s)?\d\d\d(-|\s)?\d\d\d\d)')
    mo = regex.search(item)
    print(mo.group())


# ### * (star), + (plus)

# In[ ]:


message = 'My number is 415-555-4242.'
message_1 = 'My phone number is (415)-555-4242.'
message_2 = 'My phone number is 4155554242.'

# Finds phone number using \d*.
print('This is the result of using \d* in place of \d\d\d or \d\d\d\d:')
for item in [message, message_1, message_2]:
    regex = re.compile(r'((\d*)(-|\s)?\d*(-|\s)?\d*)')
    mo = regex.search(item)
    print(mo.group())

# Finds phone number using \d+.
print('This is the result of using \d+ in place of \d\d\d or \d\d\d\d:')
for item in [message, message_1, message_2]:
    regex = re.compile(r'((\d+)(-|\s)?\d+(-|\s)?\d+)')
    mo = regex.search(item)
    print(mo.group())


# ### {} (brace)

# In[ ]:


message = 'My number is 415-555-4242.'
message_1 = 'My phone number is 415 555 4242.'
message_2 = 'My phone number is 4155554242.'

# Finds phone number using \d{3} or \d{4}
print('This is the result of using \d{3} for \d\d\d and \d{4} for \d\d\d\d:')
for item in [message, message_1, message_2]:
    regex = re.compile(r'((\d{3})(-|\s)?\d{3}(-|\s)?\d{4})')
    mo = regex.search(item)
    print(mo.group())
```

```
136
137
138 # ### .findall() and greedy matching
139 #
140 # https://pythex.org/
141
142 # In[ ]:
143
144
145 message = 'My number is 415-555-4242. My phone number is 415 555 4242. My phone number is 4155554242.'
146 message_1 = 'hahahahaha'
147
148 # Finds phone number using search(). This only returns the first number found.
149 print('This is what search() returns:')
150 regex = re.compile(r'((\d{3})(-|\s)?\d{3}(-|\s)?\d{4})')
151 mo = regex.search(message)
152 print(mo.group())
153 print('\n')
154
155 # Finds phone number using findall(). This only returns a list containing tuples of all groups.
156 print('This is what findall() returns:')
157 regex = re.compile(r'((\d{3})(-|\s)?\d{3}(-|\s)?\d{4})')
158 mo = regex.findall(message)
159 print(mo)
160 print('\n')
161
162
163 # In[ ]:
164
165
166 print('This is the message: ' + message_1 + '\n')
167 # Greedy
168 print('Regex: (ha){3,5}')
169 regex = re.compile(r'(ha){3,5}')
170 mo = regex.search(message_1)
171 print(mo.group())
172
173 # Not greedy
174 print('Regex: (ha){3,5}?')
175 regex = re.compile(r'(ha){3,5}?')
176 mo = regex.search(message_1)
177 print(mo.group())
178
179
180 # ### /d, /D, /w, /W, /s, /S
```

```
181
182 # In[ ]:
183
184
185 # \d = digit
186 # \w = letter character, cap or lower, _
187 # \s = ' '
188 # \D = not digit
189 # \W = not letter
190 # \S = not space
191
192
193 # In[ ]:
194
195
196 message = '''This is the shopping list for Christmas last year: 12 drummers, 11 pipers, 10 lords, 9 ladies,
197 8 maids, 7 swans, 6 geese, 5 rings, 4 birds, 3 hens, 2 doves, 1 partridge'''
198
199 # Returns a list of all gifts from last year Christmas.
200 print('This is the message: ' + message + '\n')
201 print('Regex: \d+\s\w+')
202 regex = re.compile(r'\d+\s\w+')
203 mo = regex.findall(message)
204 print(mo)
205
206
207 # ### [ ] (brace) and [^ ] (not)
208
209 # In[ ]:
210
211
212 message = '''Eunoia is the shortest word in English to contain all five vowels,
213 and the word quite literally means beautiful thinking.'''
214
215 print('This is the message:')
216 print(message)
217 print('\n')
218
219 # Returns a list of all vowels in message.
220 print('Regex: [AEIOUaeiou]')
221 regex = re.compile(r'[AEIOUaeiou]')
222 mo = regex.findall(message)
223 print(mo)
224
225 # Returns a list of all consonants in message. Uses
```

```
226  print('Regex: [^AEIOUaeiou\s.]): ')
227  regex = re.compile(r'[^AEIOUaeiou\s.]')
228  mo = regex.findall(message)
229  print(mo)
230
231
232  # ### ^ (caret) and $ (dollar sign)
233
234  # In[ ]:
235
236
237  message = '''Eunoia is the shortest word in English to contain all five vowels,
238  and the word quite literally means beautiful thinking.'''
239
240  print('This is the message: \n' + message + '\n')
241
242  print('Regex: ^Eunoia is ')
243  regex = re.compile(r'^Eunoia is')
244  mo = regex.findall(message)
245  print(mo)
246
247  print('Regex: Eunoia is$')
248  regex = re.compile(r'Eunoia$')
249  mo = regex.findall(message)
250  print(mo)
251
252  print('Regex: ^beautiful thinking.')
253  regex = re.compile(r'^beautiful thinking.')
254  mo = regex.findall(message)
255  print(mo)
256
257  print('Regex: beautiful thinking.$')
258  regex = re.compile(r'beautiful thinking.$')
259  mo = regex.findall(message)
260  print(mo)
261
262
263  # ### . (wildcard)
264
265  # In[ ]:
266
267
268  message = 'The cat in the hat eats on the flat mat.'
269
270  # Returns all words with -at.
```

```
print('Regex: .at')
regex = re.compile(r'(.at)')
mo = regex.findall(message)
print(mo)

print('\n')

# Returns all words with -at.
print('Or:\nRegex: \w*at')
regex = re.compile(r'\w*at')
mo = regex.findall(message)
print(mo)


# ### re.DOTALL, re.I, re.VERBOSE

# In[ ]:


message = 'The apparition of these faces in the crowd: \nPetals on a wet, black bough. \n-Ezra Pound'

print('This is the message: ' + '\n' + message + '\n')

# Finds everything, stoping at any new lines.
print('Regex: .*')
regex = re.compile(r'(.*)')
mo = regex.search(message)
print(mo.group())

print('\n')

# Finds everything.
print('Regex: .* with re.DOTALL')
regex = re.compile(r'(.*)', re.DOTALL)
mo = regex.search(message)
print(mo.group())


# In[ ]:


favorite_pokemon = "'Marshtomp', 'MarshTomp', 'MARSHTOMP', 'marshtomp'"

print('This is the message: ' + favorite_pokemon + '\n')

```

```
316   # Finds marshtomp in any combination of cases.
317   print('Regex: marshtomp with re.I')
318   regex = re.compile(r'marshtomp', re.I)
319   mo = regex.findall(favorite_pokemon)
320   print(mo)
321
322
323   # In[ ]:
324
325
326   message = '''415-555-4242, 415 555 4242, 4155554242, (415)-555-4242, 415.555.4242, (415) 555-4242, 555-4242, +13
415 555 4242'''
327
328   book = re.compile(r'''(
329       (\d{3}|\(\d{3}\))?            # area code
330       (\s|-|\.)?                    # separator
331       \d{3}                         # first 3 digits
332       (\s|-|\.)                     # separator
333       \d{4}                         # last 4 digits
334       (\s*(ext|x|ext.)\s*\d{2,5})?  # extension
335       )''', re.VERBOSE)
336
337
338   regex = re.compile(r'''
339       (\s?(\+\d+)? # international
340       (-|\s|\.)? # international separator
341       (\(?\d{3}\)?)? # area code (optional)
342       (-|\s|\.)? # separator
343       \d{3} # next 3 digits
344       (-|\s|\.)? # separator
345       \d{4} # last four digits
346       )''', re.VERBOSE)
347
348   def result_filter(lst):
349       return [item[0] for item in lst]
350
351   mo = book.findall(message)
352   print('This is the results of using the regex from the book: ')
353   print(mo)
354   print('Filtered:')
355   print(result_filter(mo))
356   print('\n')
357   result = regex.findall(message)
358   print('This is the results of using the regex we created: ')
359   print(result)
```

```
print('Filtered:')
print(result_filter(result))


# ### .sub()

# In[ ]:


message = 'Agent Romanov gave the secret documents to Agent Barton.'

# Replace all 'Agent + name' with CENSORED
regex = re.compile(r'Agent\s\w+')
mo = regex.sub('CENSORED', message)
print(mo)


# In[ ]:
```