ANS=1

In [1]:
```
import re
text = 'Python Exercise, PHP exercises.'
print(re.sub('[ ,.]',':',text))
```
```
Python:Exercise::PHP:exercises:
```

ANS=2

In [4]:
```
import re
text='Python Exercise,PHP exercise'
pattern=re.findall('[aeA-E]\w+',text)
print(pattern)
```
```
['Exercise', 'exercise']
```

ANS=3

In [15]:
```
import re
text='the information collected in all such cases is called data.'
pattern=re.compile(r'(\b\w{4,}\b)')
for match in pattern.finditer(text):
    print(match.group(1))
```
```
information
collected
such
cases
called
data
```

ANS=4

In [17]:
```
import re
text='the information collected in all such cases is called data.'
pattern=re.compile(r'(\b\w{3,5}\b)')
for match in pattern.finditer(text):
    print(match.group(1))
```
```
the
all
such
cases
data
```

ANS=6

In [1]:
```
import re

def remove_parentheses(strings):
  pattern = re.compile(r'\([^)]*\)')

  modified_strings = []
  for string in strings:
    modified_string = re.sub(pattern, '', string)
    modified_strings.append(modified_string)

  return modified_strings


sample_text = ["example (.com)", "hr@fliprobo (.com)", "github (.com)", "Hello (Data Science World)", "Data (Scientist)"]
result = remove_parentheses(sample_text)
print(result)
```
```
['example ', 'hr@fliprobo ', 'github ', 'Hello ', 'Data ']
```

ANS=7

In [33]:
```
import re
text = "ImportanceOfRegularExpressionsInPython"
print(re.findall("[A-Z][^A-Z]*", text))
```
```
['Importance', 'Of', 'Regular', 'Expressions', 'In', 'Python']
```

ANS=8

In [44]:
```
import re
Text ="RegularExpression1IsAn2ImportantTopic3InPython"
y=re.sub(r"(\w)([0-9])", r"\1 \2", Text)
print(y)
```
```
RegularExpression 1IsAn 2ImportantTopic 3InPython
```

ANS=9

In [47]:
```
import re
Text =  "RegularExpression1IsAn2ImportantTopic3InPython"
y=re.sub(r"(\w)([A-Z])", r"\1 \2", Text)
print(y)
```
```
Regular Expression1 Is An2 Important Topic3 In Python
```

ANS=11

In [49]:
```
import re
def match_function(text):
    pattern="^[a-zA-Z0-9_]*$"
    if re.search(pattern,text):
        return "match"
    else:
        return "not match"
print(match_function("he is a gud boy"))
print(match_function("He_Play_2"))
```
```
not match
match
```

ANS=12

In [60]:
```
import re
def match_spe_num(text):
    pattern = re.compile(r"^8")
    if pattern.match(text):
        return True
    else:
        return False
print(match_spe_num('8-2369745'))
print(match_spe_num('6-1236528'))
```
```
True
False
```

ANS=13

In [61]:
```
import re
ip = "216.08.094.196"
text = re.sub('\.[0]*', '.', ip)
print(text)
```
```
216.8.94.196
```

ANS=14

In [2]:
```
import re

text = "On August 15th 1947 that India was declared independent from British colonialism, and the reins of control were handed over to the leaders of the Country."

pattern = r"\b([A-Z][a-z]+ \d{1,2}(?:st|nd|rd|th)? \d{4})\b"

matches = re.findall(pattern, text)
date_string = matches[0] if matches else None

print(date_string)
```
```
August 15th 1947
```

ANS=15

In [69]:
```
import re
pattern = [ 'fox', 'dog', 'horse' ]
text = 'The quick brown fox jumps over the lazy dog.'
for i in pattern:
    print('Searching for "%s" in "%s" ' % (i, text),)
    if re.search(i,  text):
        print('Matched')
    else:
        print('Not Matched')
```
```
Searching for "fox" in "The quick brown fox jumps over the lazy dog."
Matched
Searching for "dog" in "The quick brown fox jumps over the lazy dog."
Matched
Searching for "horse" in "The quick brown fox jumps over the lazy dog."
Not Matched
```

ANS=16

In [76]:
```
import re
pattern = 'fox'
text = 'The quick brown fox jumps over the lazy dog.'
match = re.search(pattern, text)
s = match.start()
e = match.end()
print('Found "%s" in "%s" from %d to %d' % \
     (match.re.pattern, match.string, s, e))
```
```
Found "fox" in "The quick brown fox jumps over the lazy dog." from 16 to 19
```

ANS=17

In [79]:
```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.findall(pattern, text):
    print('Found "%s"' % match)
```
```
Found "exercises"
Found "exercises"
Found "exercises"
```

ANS=18

In [80]:
```
import re
text = 'Python exercises, PHP exercises, C# exercises'
pattern = 'exercises'
for match in re.finditer(pattern, text):
    s = match.start()
    e = match.end()
    print('Found "%s" at %d:%d' % (text[s:e], s, e))
```
```
Found "exercises" at 7:16
Found "exercises" at 22:31
Found "exercises" at 36:45
```

ANS=19

In [81]:
```
import re
def change_date_format(dt):
        return re.sub(r'(\d{4})-(\d{1,2})-(\d{1,2})', '\\3-\\2-\\1', dt)
dt1 = "2026-01-02"
print("Original date in YYY-MM-DD Format: ",dt1)
print("New date in DD-MM-YYYY Format: ",change_date_format(dt1))
```
```
Original date in YYY-MM-DD Format:  2026-01-02
New date in DD-MM-YYYY Format:  02-01-2026
```

ANS=20

In [85]:
```
def is_decimal(num):
    import re
    dnum = re.compile(r"""^[0-9]+(\.[0-9]{1,2})?$""")
    result = dnum.search(num)
    return bool(result)

print(is_decimal('123.11'))
print(is_decimal('123.1'))
print(is_decimal('123'))
print(is_decimal('0.21'))

print(is_decimal('123.1214'))
print(is_decimal('3.124587'))
print(is_decimal('e666.86'))
```
```
True
True
True
True
False
False
False
```

ANS=21

In [6]:
```
import re
text = "Rohit sharma scored 43 centuries and 91 half centuries in his cricket carrier"

for i in re.finditer("\d+", text):
    print(i.group(0))
    print("position=", i.start())
```
```
43
position= 20
91
position= 37
```

ANS=22

In [7]:
```
import re
text='My marks in each semester are: 947, 896, 926, 524, 734, 950, 642'
pattern = re.findall('\d+', text)
pattern = map(int, pattern)
print("Max_value:",max(pattern))
```
```
Max_value: 950
```

ANS=23

In [5]:
```
import re
Text =  "RegularExpressionIsAnImportantTopicInPython"
y=re.sub(r"(\w)([A-Z])", r"\1 \2", Text)
print(y)
```
```
Regular Expression Is An Important Topic In Python
```

ANS=24

In [8]:
```
import re
def ufl_match(text):
        pattern = '[A-Z]+[a-z]+$'
        if re.search(pattern, text):
                return 'match'
        else:
                return('not match')
print(ufl_match("Amesterdam"))
print(ufl_match("NEWYORK"))
print(ufl_match("python"))
print(ufl_match("Paris"))
print(ufl_match("California"))
print(ufl_match("VETICANCITY"))
```
```
match
not match
not match
match
match
not match
```

ANS=25

In [1]:
```
import re


def consecutiveWords(text):
    pattern=r'\b(\w+)(?:\W+\1\b)+'
    return re.sub(pattern, r'\1', text)
y="Hello Hello world world"
print(consecutiveWords(y))
```
```
Hello world
```

ANS=26

In [2]:
```
import re
pattern = '[a-zA-z0-9]$'
def check(text):
    if(re.search(pattern, text)):
        print("end with alpha")

    else:
        print("not end with alpha")

if __name__ == '__main__' :

    x = "malaysia3261"

    check(x)

    y = "robothook."
    check(y)
```
```
end with alpha
not end with alpha
```

ANS=27

In [1]:
```
import re

def remove_hashtags(text):
  hashtags = re.findall(r'#\w+', text)
  return hashtags

sample_text = 'RT @kapil_kausik: #Doltiwal I mean #xyzabc is "hurt" by #Demonetization as the same has rendered USELESS <ed><U+00A0><U+00BD><ed><U+00B1><U+0089> "acquired funds" No wo'
hashtags = remove_hashtags(sample_text)
print(hashtags)
```
```
['#Doltiwal', '#xyzabc', '#Demonetization']
```

ANS=28

In [4]:
```
import re

sample_text = "@Jags123456 Bharat band on 28??<ed><U+00A0><U+00BD><ed><U+00B8><U+0082>Those who are protesting #demonetization are all different party leaders"

modified_text = re.sub(r"<U\+\w{4}>", "", sample_text)

print(modified_text)
```
```
@Jags123456 Bharat band on 28??<ed><ed>Those who are protesting #demonetization are all different party leaders
```

ANS=29

In [5]:
```
import re
text="Ron was born on 12-09-1992 and he was admitted to school 15-12-1999"
pattern=r'\d{2}-\d{2}-\d{4}'
dates = re.findall(pattern, text)
for date in dates:
  print(date)
```
```
12-09-1992
15-12-1999
```

ANS=30

In [6]:
```
import re

def remove_words(string):
  pattern = re.compile(r'\b\w{2,4}\b')
  modified_string = re.sub(pattern, '', string)
  return modified_string


sample_text = "The following example creates an ArrayList with a capacity of 50 elements. 4 elements are then added to the ArrayList and the ArrayList is trimmed accordingly."
result = remove_words(sample_text)
print(result)
```
```
 following example creates  ArrayList  a capacity   elements. 4 elements   added   ArrayList   ArrayList  trimmed accordingly.
```

In [ ]: