Formal Languages and Compilers

Lab4

Regular Expressions in Python

- **Regular expression (regex):** sequence of characters that indicates a search pattern
  - Use: define filters, email format checker, password pattern matching, search engines, search & replace text

- **re**: built-in Python in module which is used for working with regular expressions
- To use it you need to import it: *import re*

Functions

- search(): returns a match object (=object containing information about the search and the result) if there is a match anywhere in the string
- match(): returns a corresponding match object if 0 or more character at the beginning of a string match this regular expression; if the string does not match the pattern, it returns None
- fullmatch(): returns a corresponding match object if the whole string matches the regular expression; if not, it returns None
- findall(): returns a list containing all matches
- split(): returns a list where the string has been split at each match
- sub(): replaces one or many matches with a string

Metacharacters

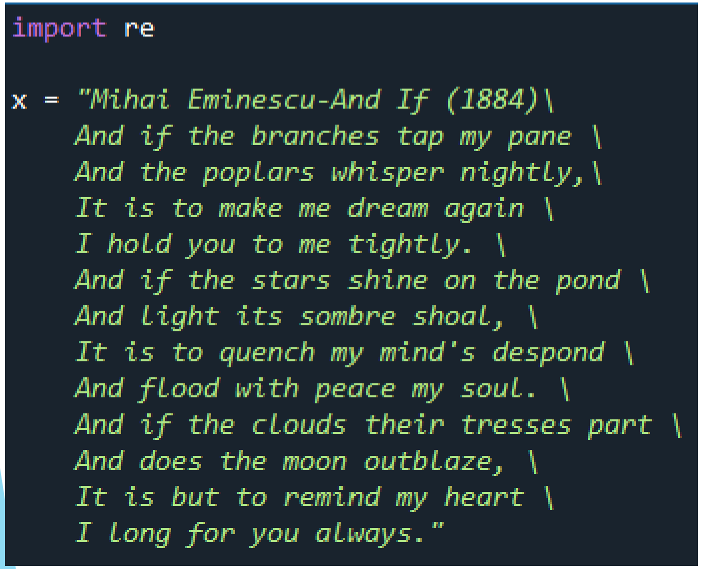
Character	Description	Example
[]	A set of characters	“[b-i]”
\	Signals a special sequence	“\d”
.	Any character (except newline)	“st…nt”
^	Starts with	“^Student”
$	Ends with	“student$”
*	Zero or more occurrences	“st.*nt”
+	One or more occurrences	“st.+nt”
?	Zero or one occurrences	“stud.?nt”
{}	Exactly the specified number of occurrences	“st.{3}nt”
|	Either or	“student|professor”
()	Capture and group	

Special Sequences

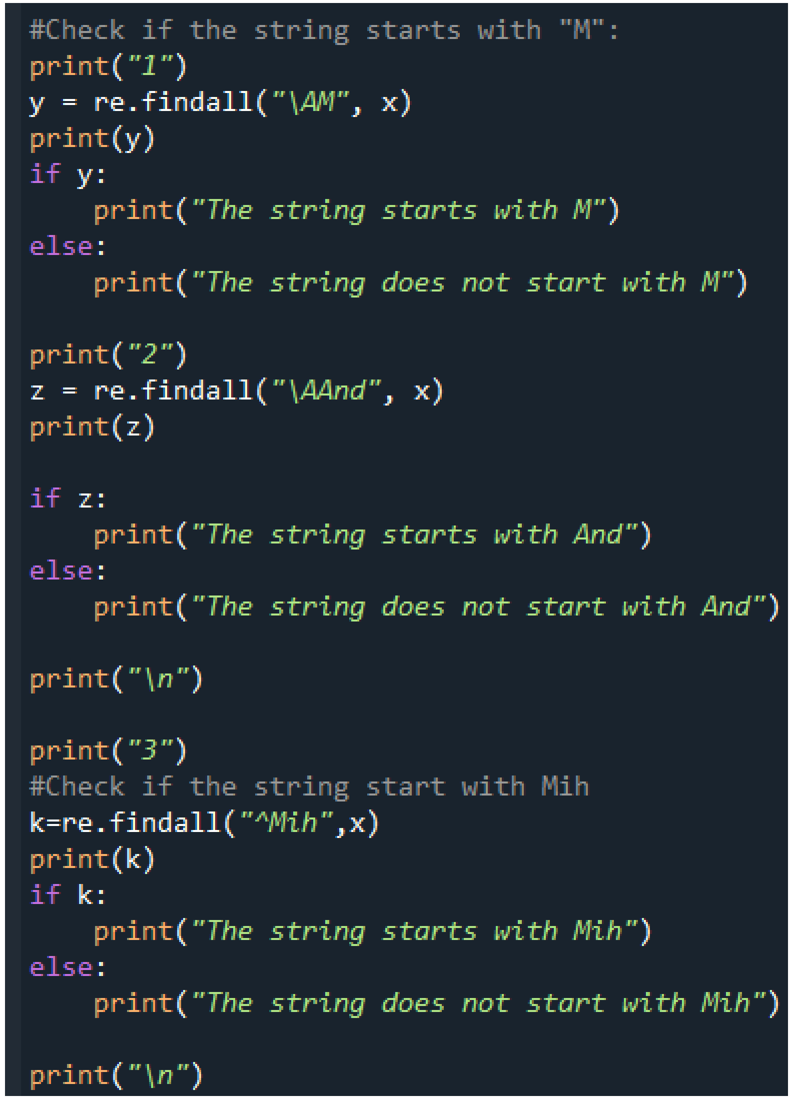
Character	Description	Example
\A	Returns a match if the specified characters are at the beginning of the string	“\AAnd”
\b	Returns a match where the specified characters are at the beginning or at the end of a word (“r” is used to make sure the string is being treated as “raw string”)	r”\bAnd” r”nd\b”
\B	Returns a match where the specified characters are present, but not at the beginning or at the end of a word	r”\Bmin” r”min\B”
\d	Returns a match where the string contains digits	“\d”
\D	Returns a match where the string does not contain digits	“\D”
\s	Returns a match where the string contains a white space character	“\s”
\S	Returns a match where the string does not contain a white space character	“\S”
\w	Returns a match where the string contains any word characters (a-Z, 0-9, _)	“\w”
\W	Returns a match where the string does not contain any word characters	“\W”
\Z	Returns a match if the specified characters are at the end of the string	“Student\Z”

Examples

Import re module
Assign string x with a value

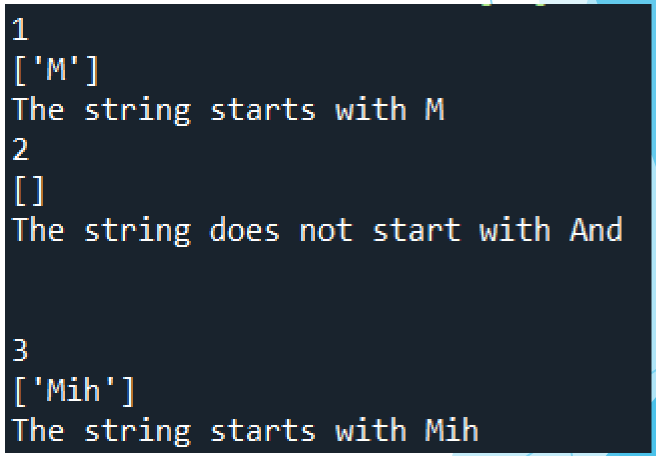


```python
import re

x = "Mihai Eminescu-And If (1884)\
    And if the branches tap my pane \
    And the poplars whisper nightly,\
    It is to make me dream again \
    I hold you to me tightly. \
    And if the stars shine on the pond \
    And light its sombre shoal, \
    It is to quench my mind's despond \
    And flood with peace my soul. \
    And if the clouds their tresses part \
    And does the moon outblaze, \
    It is but to remind my heart \
    I long for you always."
```



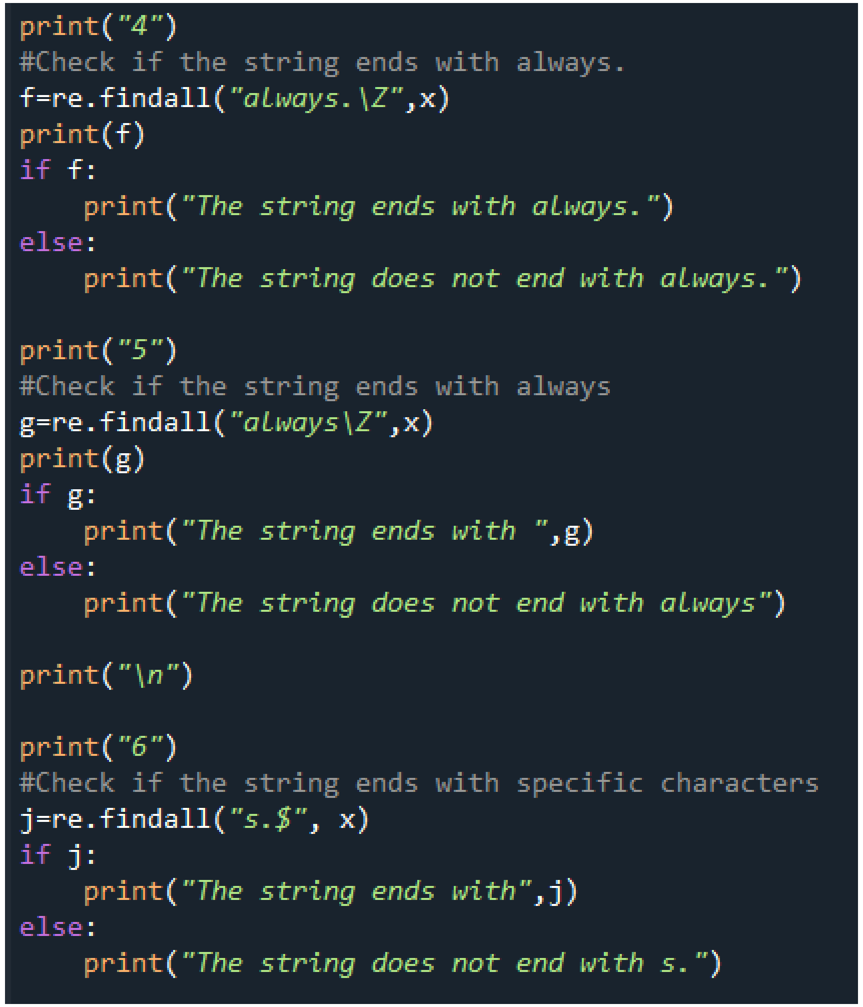
```python
#Check if the string starts with "M":
print("1")
y = re.findall("\AM", x)
print(y)
if y:
    print("The string starts with M")
else:
    print("The string does not start with M")

print("2")
z = re.findall("\AAnd", x)
print(z)

if z:
    print("The string starts with And")
else:
    print("The string does not start with And")

print("\n")

print("3")
#Check if the string start with Mih
k=re.findall("^Mih",x)
print(k)
if k:
    print("The string starts with Mih")
else:
    print("The string does not start with Mih")

print("\n")
```

Output

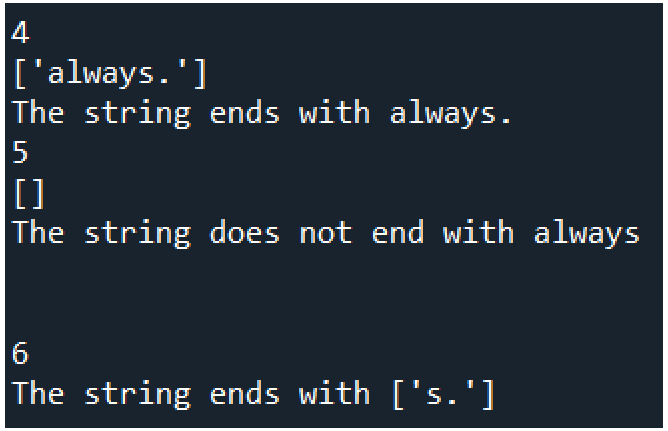


```
1
['M']
The string starts with M
2
[]
The string does not start with And


3
['Mih']
The string starts with Mih
```

Examples

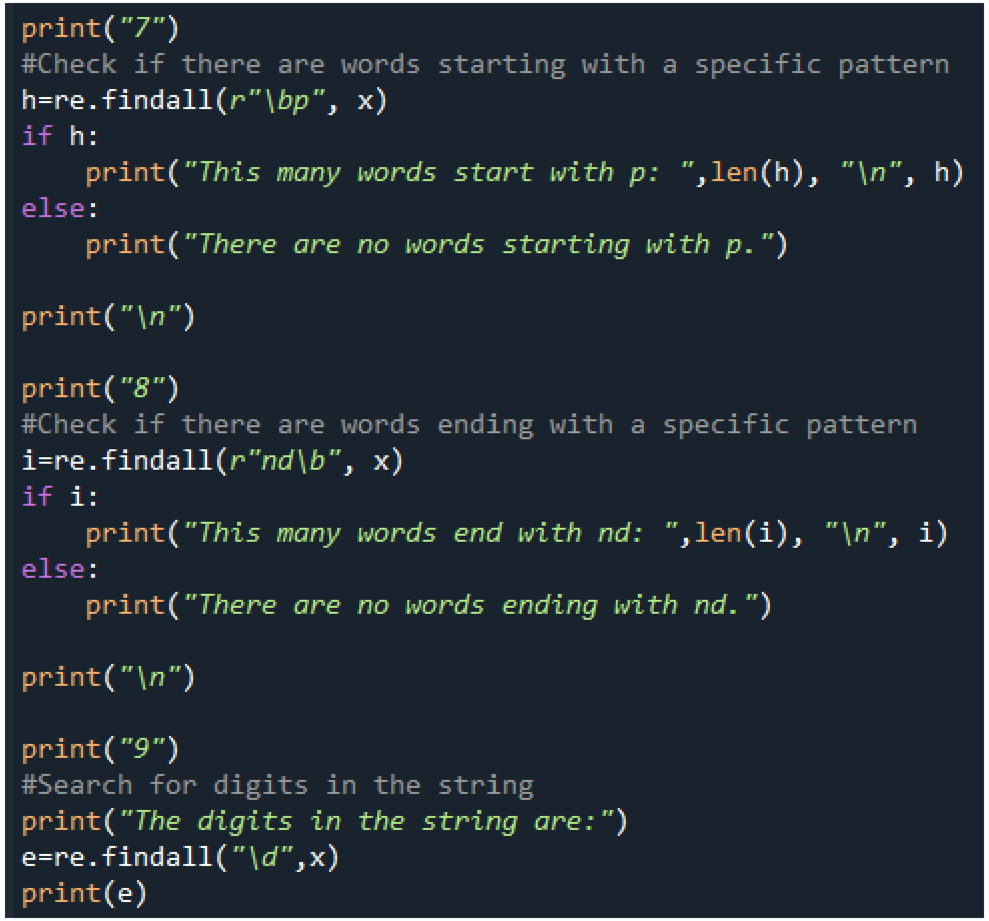


```
print("4")
#Check if the string ends with always.
f=re.findall("always.\Z",x)
print(f)
if f:
    print("The string ends with always.")
else:
    print("The string does not end with always.")

print("5")
#Check if the string ends with always
g=re.findall("always\Z",x)
print(g)
if g:
    print("The string ends with ",g)
else:
    print("The string does not end with always")

print("\n")

print("6")
#Check if the string ends with specific characters
j=re.findall("s.$", x)
if j:
    print("The string ends with",j)
else:
    print("The string does not end with s.")
```

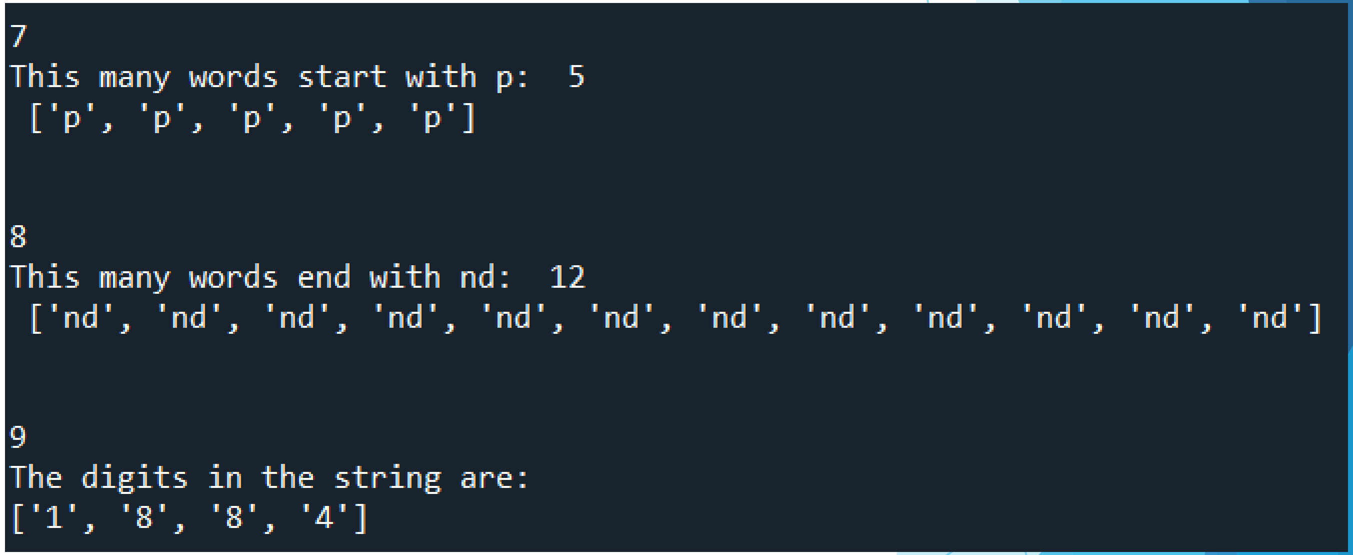


```
4
['always.']
The string ends with always.
5
[]
The string does not end with always


6
The string ends with ['s.']
```

Examples

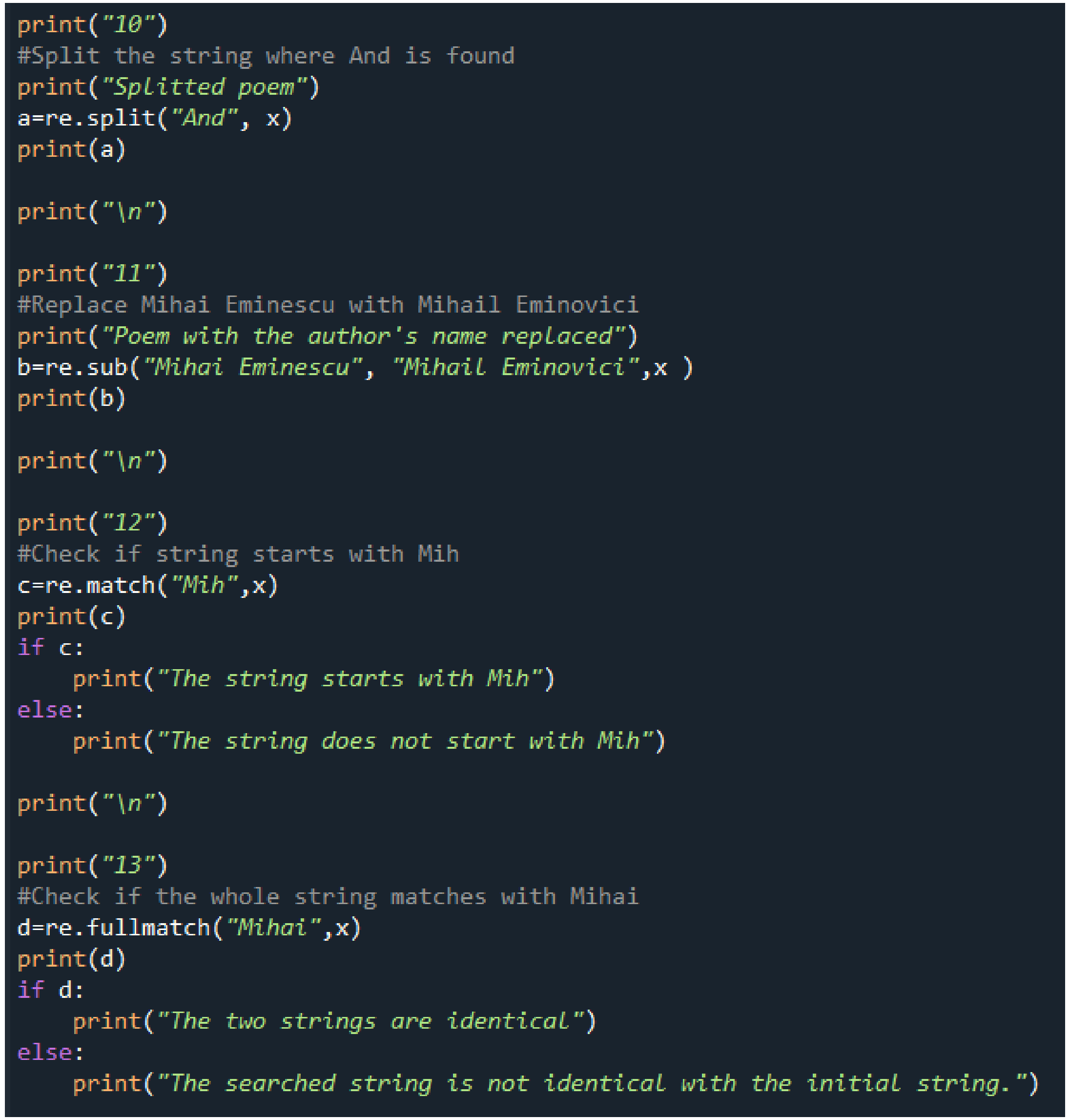


```python
print(”7”)
#Check if there are words starting with a specific pattern
h=re.findall(r”\bp”, x)
if h:
    print(”This many words start with p: ”,len(h), ”\n”, h)
else:
    print(”There are no words starting with p.”)

print(”\n”)

print(”8”)
#Check if there are words ending with a specific pattern
i=re.findall(r”nd\b”, x)
if i:
    print(”This many words end with nd: ”,len(i), ”\n”, i)
else:
    print(”There are no words ending with nd.”)

print(”\n”)

print(”9”)
#Search for digits in the string
print(”The digits in the string are:”)
e=re.findall(”\d”,x)
print(e)
```

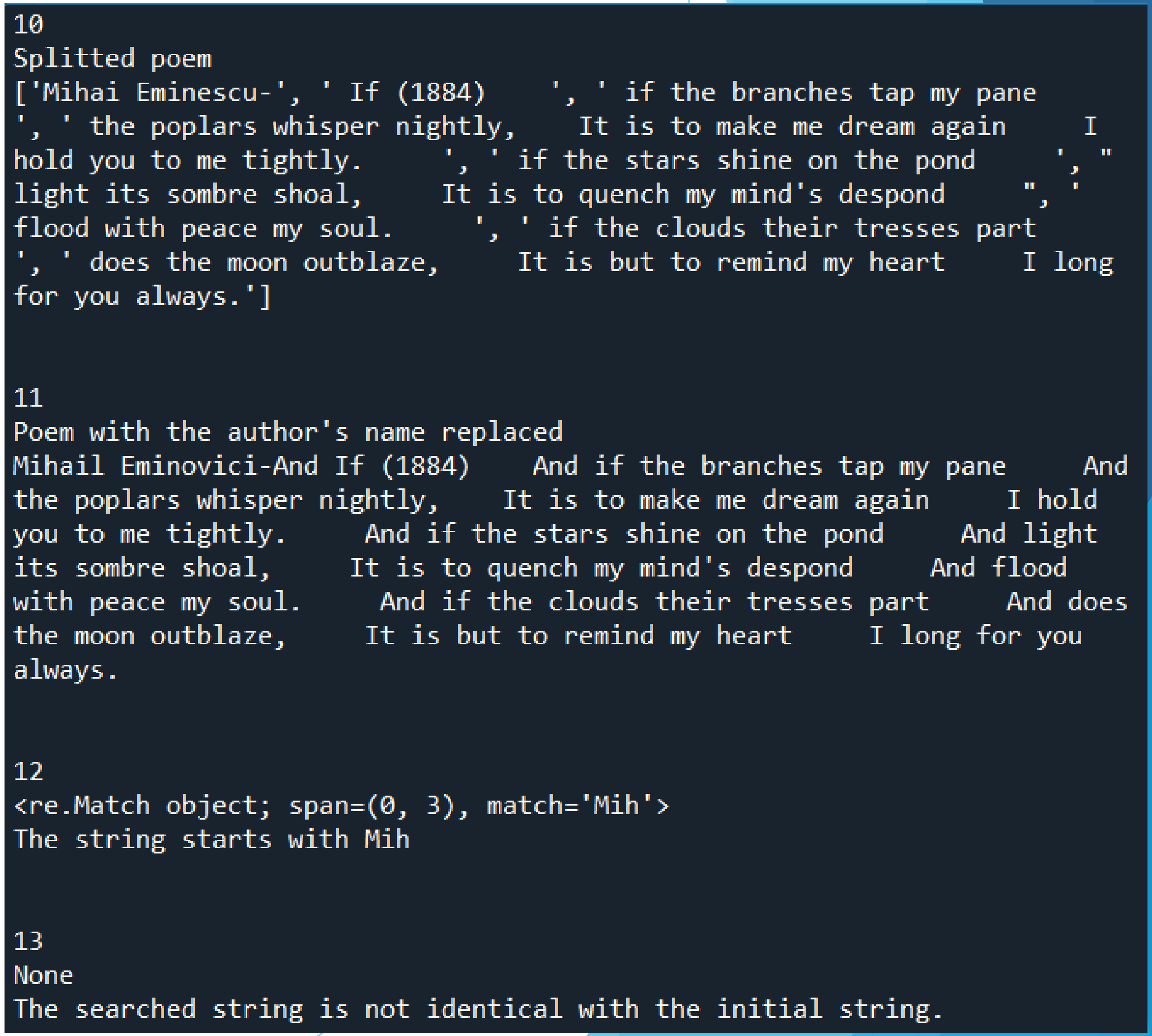


```
7
This many words start with p:  5
 ['p', 'p', 'p', 'p', 'p']


8
This many words end with nd:  12
 ['nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd', 'nd']


9
The digits in the string are:
['1', '8', '8', '4']
```

Examples

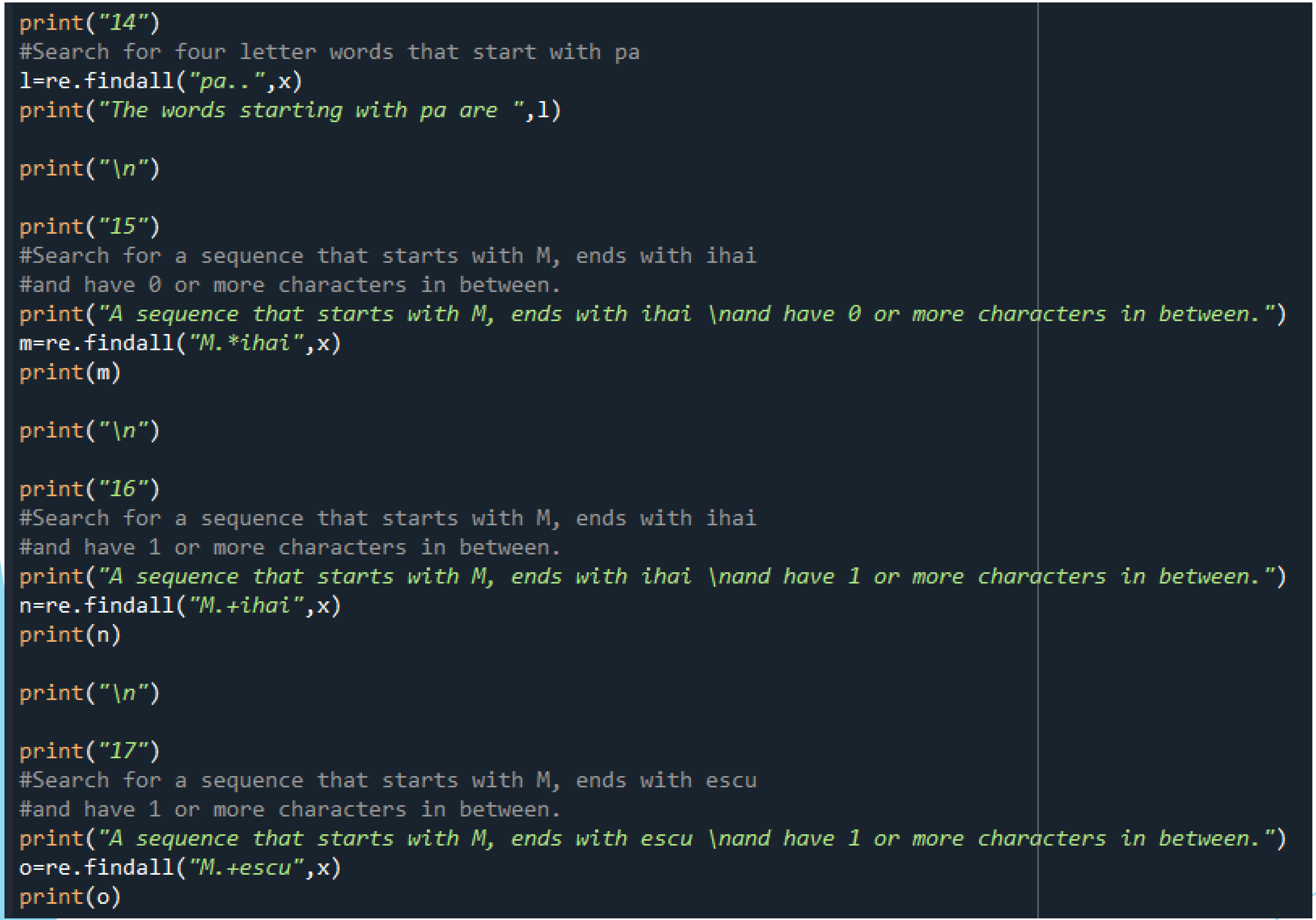


```
print("10")
#Split the string where And is found
print("Splitted poem")
a=re.split("And", x)
print(a)

print("\n")

print("11")
#Replace Mihai Eminescu with Mihail Eminovici
print("Poem with the author's name replaced")
b=re.sub("Mihai Eminescu", "Mihail Eminovici",x )
print(b)

print("\n")

print("12")
#Check if string starts with Mih
c=re.match("Mih",x)
print(c)
if c:
    print("The string starts with Mih")
else:
    print("The string does not start with Mih")

print("\n")

print("13")
#Check if the whole string matches with Mihai
d=re.fullmatch("Mihai",x)
print(d)
if d:
    print("The two strings are identical")
else:
    print("The searched string is not identical with the initial string.")
```

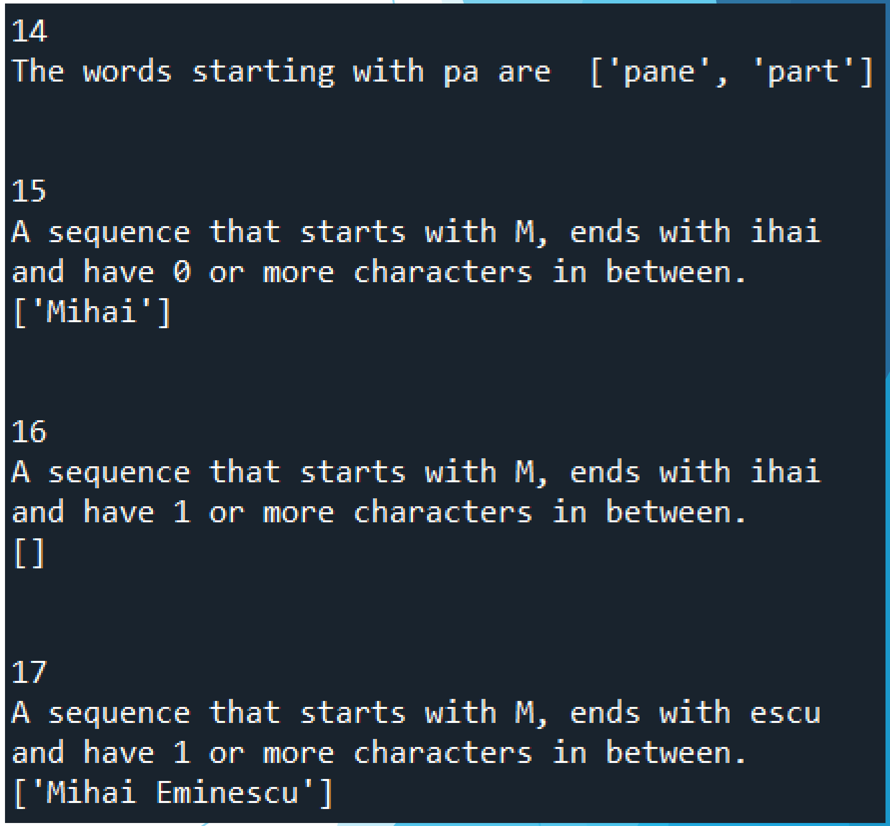


```
10
Splitted poem
['Mihai Eminescu-', ' If (1884)    ', ' if the branches tap my pane
', ' the poplars whisper nightly,    It is to make me dream again    I
hold you to me tightly.    ', ' if the stars shine on the pond    ', "
light its sombre shoal,    It is to quench my mind's despond    ", '
flood with peace my soul.    ', ' if the clouds their tresses part
', ' does the moon outblaze,    It is but to remind my heart    I long
for you always.']


11
Poem with the author's name replaced
Mihail Eminovici-And If (1884)    And if the branches tap my pane    And
the poplars whisper nightly,    It is to make me dream again    I hold
you to me tightly.    And if the stars shine on the pond    And light
its sombre shoal,    It is to quench my mind's despond    And flood
with peace my soul.    And if the clouds their tresses part    And does
the moon outblaze,    It is but to remind my heart    I long for you
always.


12
<re.Match object; span=(0, 3), match='Mih'>
The string starts with Mih


13
None
The searched string is not identical with the initial string.
```

Examples

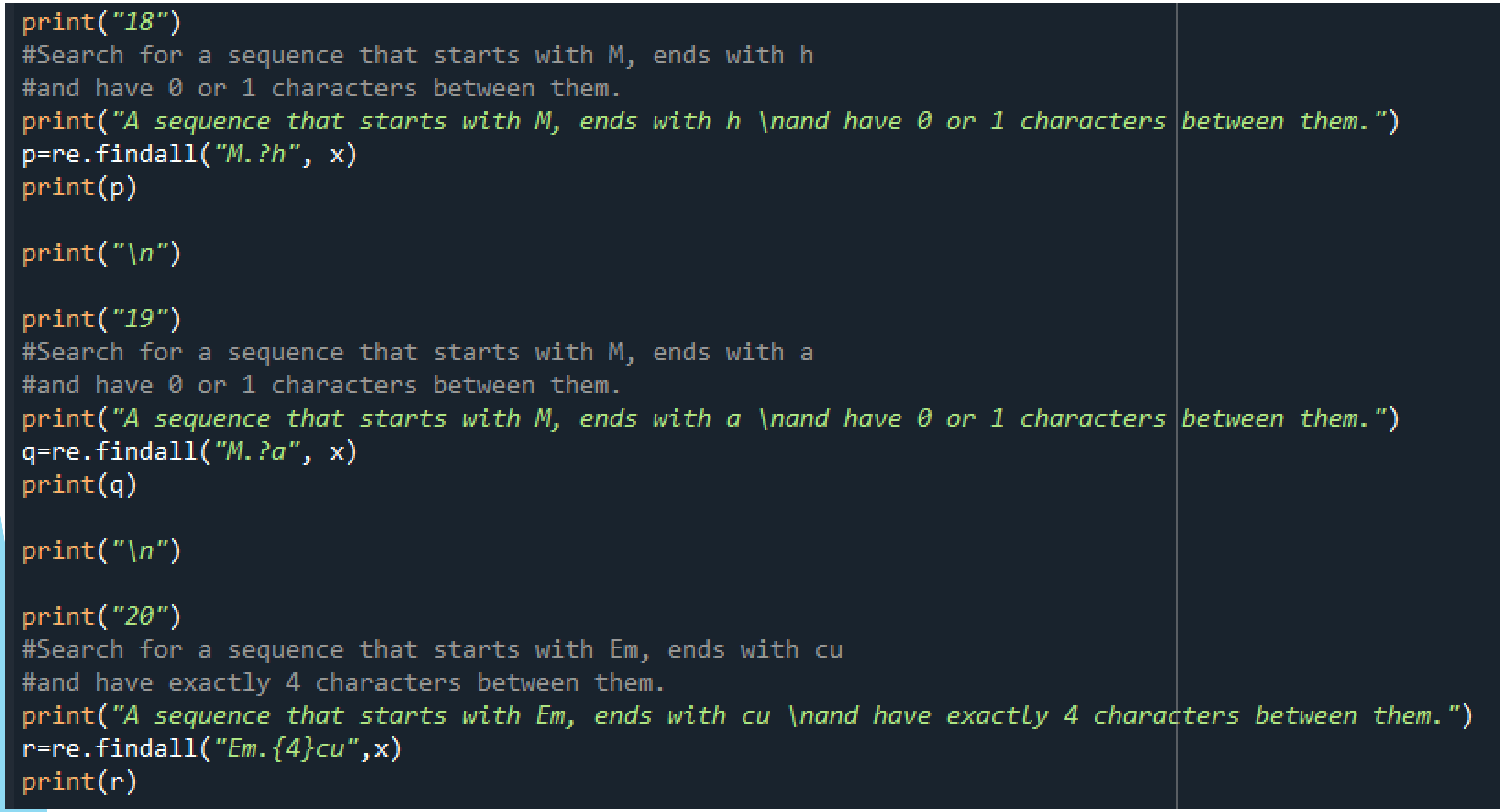


```
print("14")
#Search for four letter words that start with pa
l=re.findall("pa..",x)
print("The words starting with pa are ",l)

print("\n")

print("15")
#Search for a sequence that starts with M, ends with ihai
#and have 0 or more characters in between.
print("A sequence that starts with M, ends with ihai \nand have 0 or more characters in between.")
m=re.findall("M.*ihai",x)
print(m)

print("\n")

print("16")
#Search for a sequence that starts with M, ends with ihai
#and have 1 or more characters in between.
print("A sequence that starts with M, ends with ihai \nand have 1 or more characters in between.")
n=re.findall("M.+ihai",x)
print(n)

print("\n")

print("17")
#Search for a sequence that starts with M, ends with escu
#and have 1 or more characters in between.
print("A sequence that starts with M, ends with escu \nand have 1 or more characters in between.")
o=re.findall("M.+escu",x)
print(o)
```

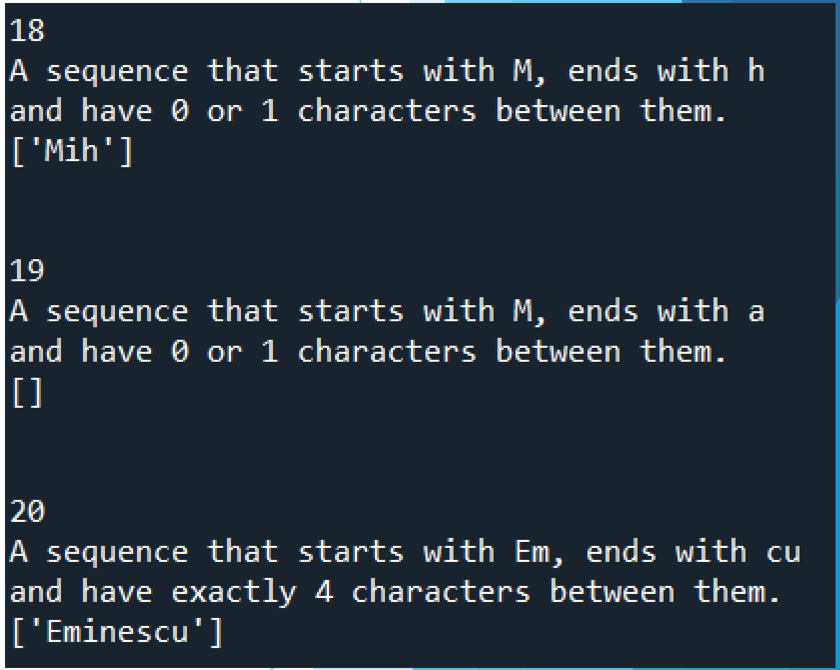


```
14
The words starting with pa are  ['pane', 'part']


15
A sequence that starts with M, ends with ihai
and have 0 or more characters in between.
['Mihai']


16
A sequence that starts with M, ends with ihai
and have 1 or more characters in between.
[]


17
A sequence that starts with M, ends with escu
and have 1 or more characters in between.
['Mihai Eminescu']
```

Examples

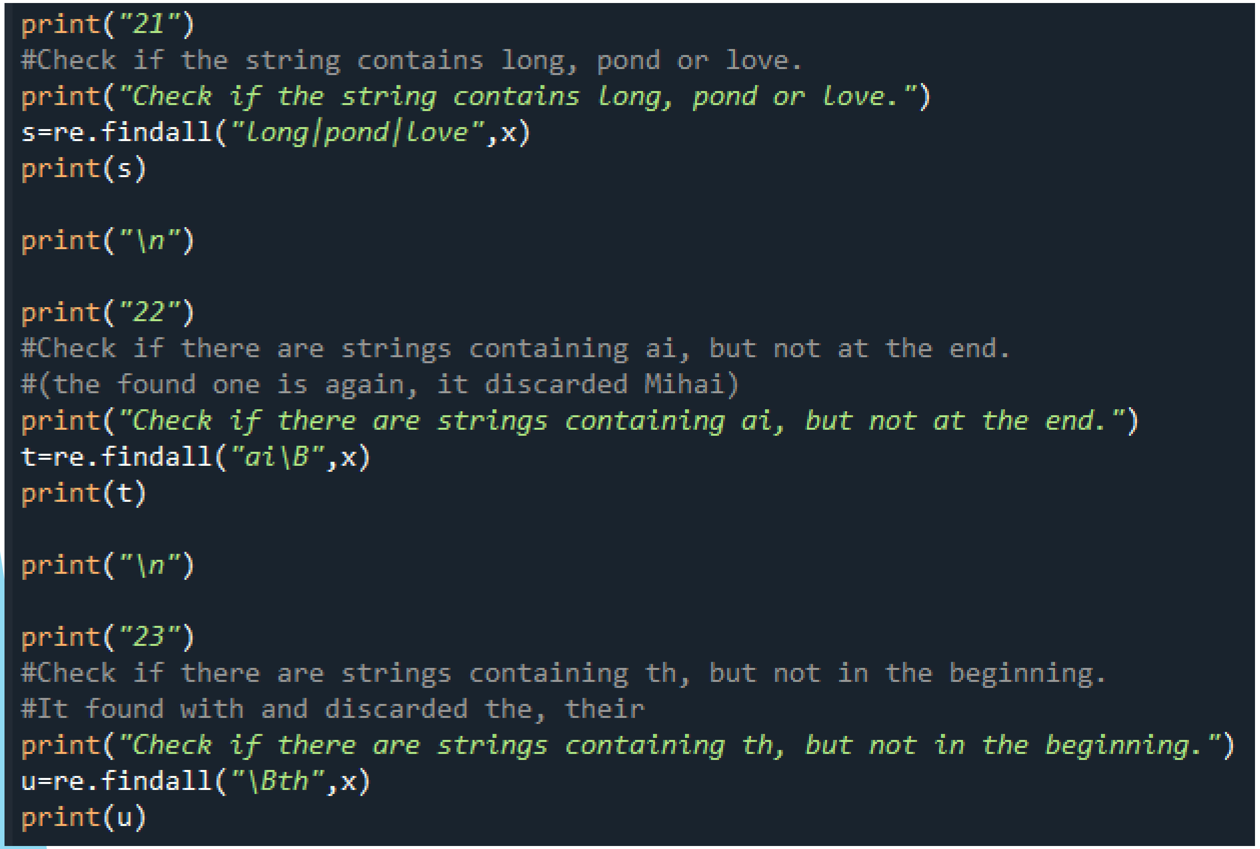


```python
print("18")
#Search for a sequence that starts with M, ends with h
#and have 0 or 1 characters between them.
print("A sequence that starts with M, ends with h \nand have 0 or 1 characters between them.")
p=re.findall("M.?h", x)
print(p)

print("\n")

print("19")
#Search for a sequence that starts with M, ends with a
#and have 0 or 1 characters between them.
print("A sequence that starts with M, ends with a \nand have 0 or 1 characters between them.")
q=re.findall("M.?a", x)
print(q)

print("\n")

print("20")
#Search for a sequence that starts with Em, ends with cu
#and have exactly 4 characters between them.
print("A sequence that starts with Em, ends with cu \nand have exactly 4 characters between them.")
r=re.findall("Em.{4}cu",x)
print(r)
```

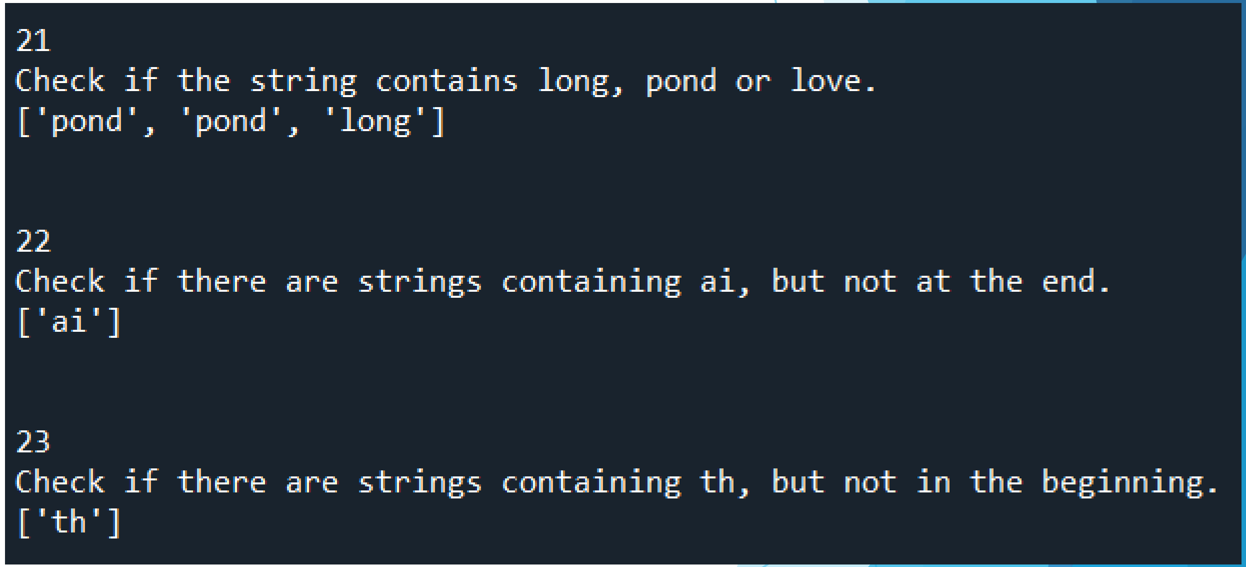


```
18
A sequence that starts with M, ends with h
and have 0 or 1 characters between them.
['Mih']


19
A sequence that starts with M, ends with a
and have 0 or 1 characters between them.
[]


20
A sequence that starts with Em, ends with cu
and have exactly 4 characters between them.
['Eminescu']
```

Examples



```python
print("21")
#Check if the string contains long, pond or love.
print("Check if the string contains long, pond or love.")
s=re.findall("long|pond|love",x)
print(s)

print("\n")

print("22")
#Check if there are strings containing ai, but not at the end.
#(the found one is again, it discarded Mihai)
print("Check if there are strings containing ai, but not at the end.")
t=re.findall("ai\B",x)
print(t)

print("\n")

print("23")
#Check if there are strings containing th, but not in the beginning.
#It found with and discarded the, their
print("Check if there are strings containing th, but not in the beginning.")
u=re.findall("\Bth",x)
print(u)
```



```
21
Check if the string contains long, pond or love.
['pond', 'pond', 'long']


22
Check if there are strings containing ai, but not at the end.
['ai']


23
Check if there are strings containing th, but not in the beginning.
['th']
```

Sets

- =set of characters with a special meaning, inside []

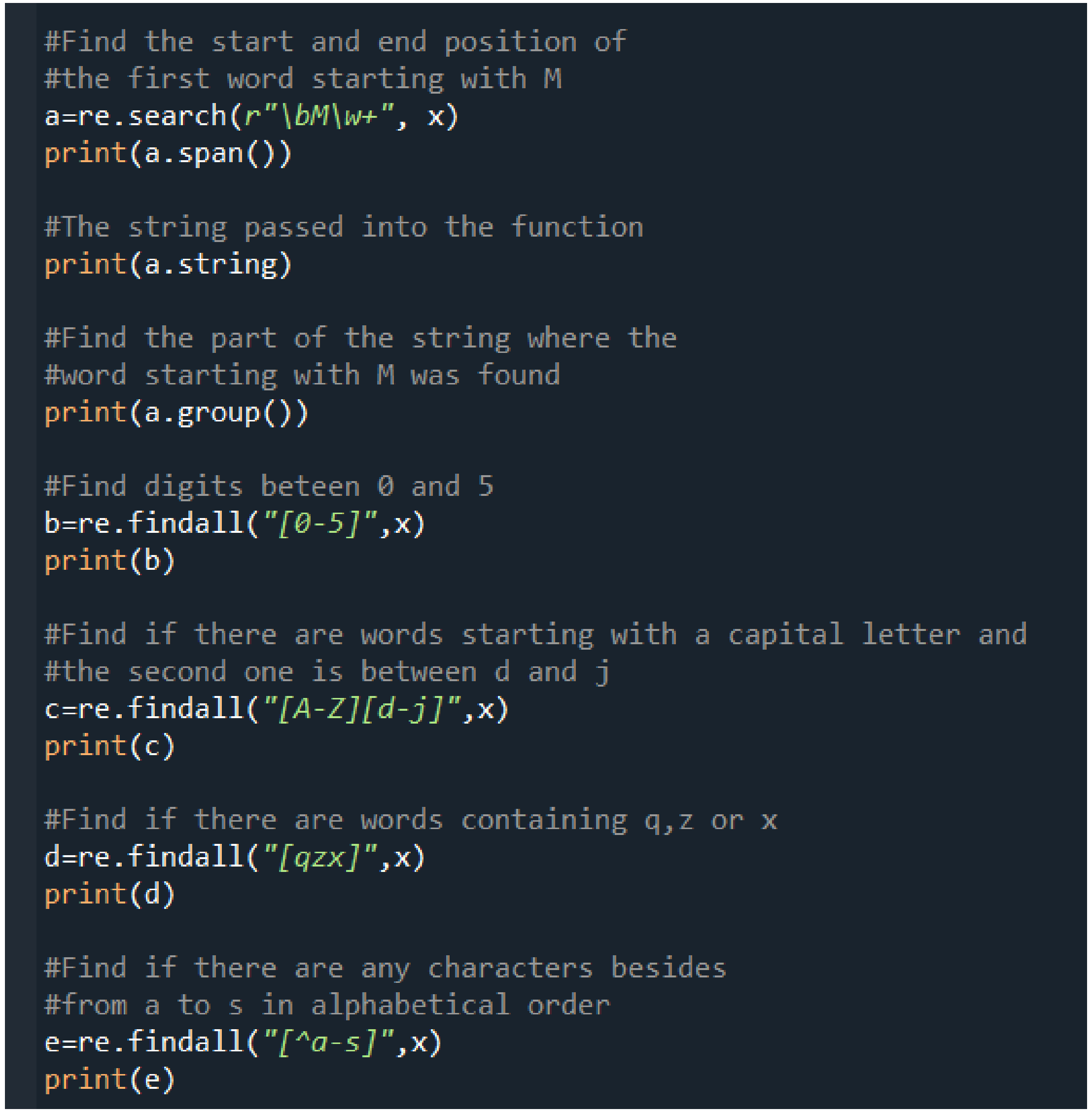
Set	Description
[abc]	Returns a match where the specified characters (a,b,c) are found
[b-l]	Returns a match where lowercase alphabetical characters between b and l are found
[^abc]	Returns a match for any character except the ones specified (except a,b,c)
[123]	Returns a match where the specified digits (1,2,3) are found
[3-8]	Returns a match for any digit between 3 and 8
[0-3][0-9]	Returns a match for any two-digits number between 00 and 39
[a-zA-Z]	Returns a match for any alphabetical character from a to z and A to Z (the whole alphabet, lowercase and uppercase)
[.]	Returns a match where the specified character (.) is found. In sets, +,*,.,|,(),$,{} have no special meaning.

Match objects

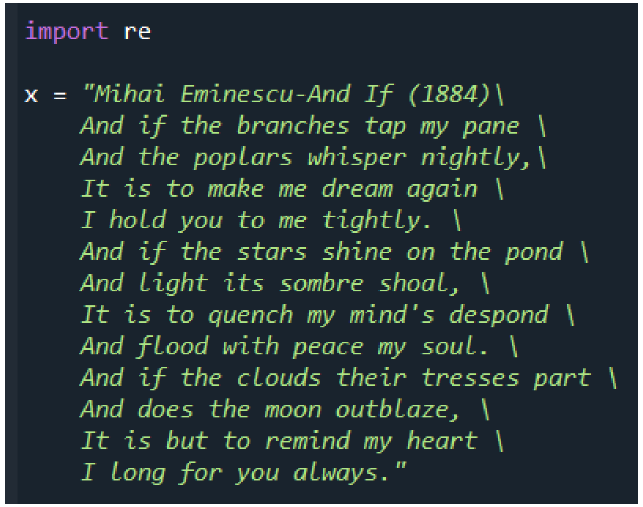
- =object containing information about the search and result (if no match, None will be returned)

- span() – returns a tuple containing the start and end positions of the match
- string – returns the string passed into the function
- group() – returns the part of the string where there was a match
- groups() – returns a tuple containing all the group matches
- compile() – compile a regular expression pattern into a regular expression object (efficient if the regex is used multiple times)

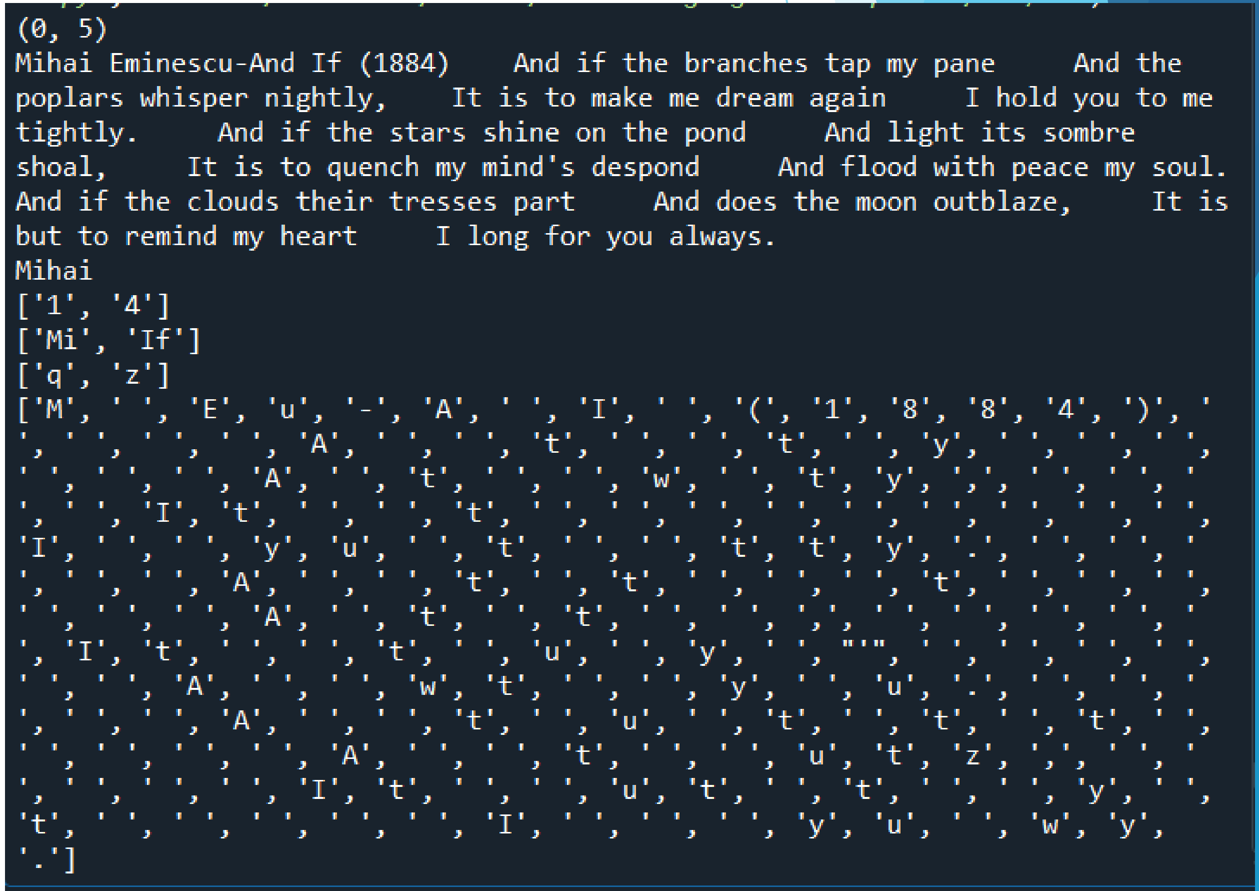
Examples



```
#Find the start and end position of
#the first word starting with M
a=re.search(r"\bM\w+", x)
print(a.span())

#The string passed into the function
print(a.string)

#Find the part of the string where the
#word starting with M was found
print(a.group())

#Find digits beteen 0 and 5
b=re.findall("[0-5]",x)
print(b)

#Find if there are words starting with a capital letter and
#the second one is between d and j
c=re.findall("[A-Z][d-j]",x)
print(c)

#Find if there are words containing q,z or x
d=re.findall("[qzx]",x)
print(d)

#Find if there are any characters besides
#from a to s in alphabetical order
e=re.findall("[^a-s]",x)
print(e)
```

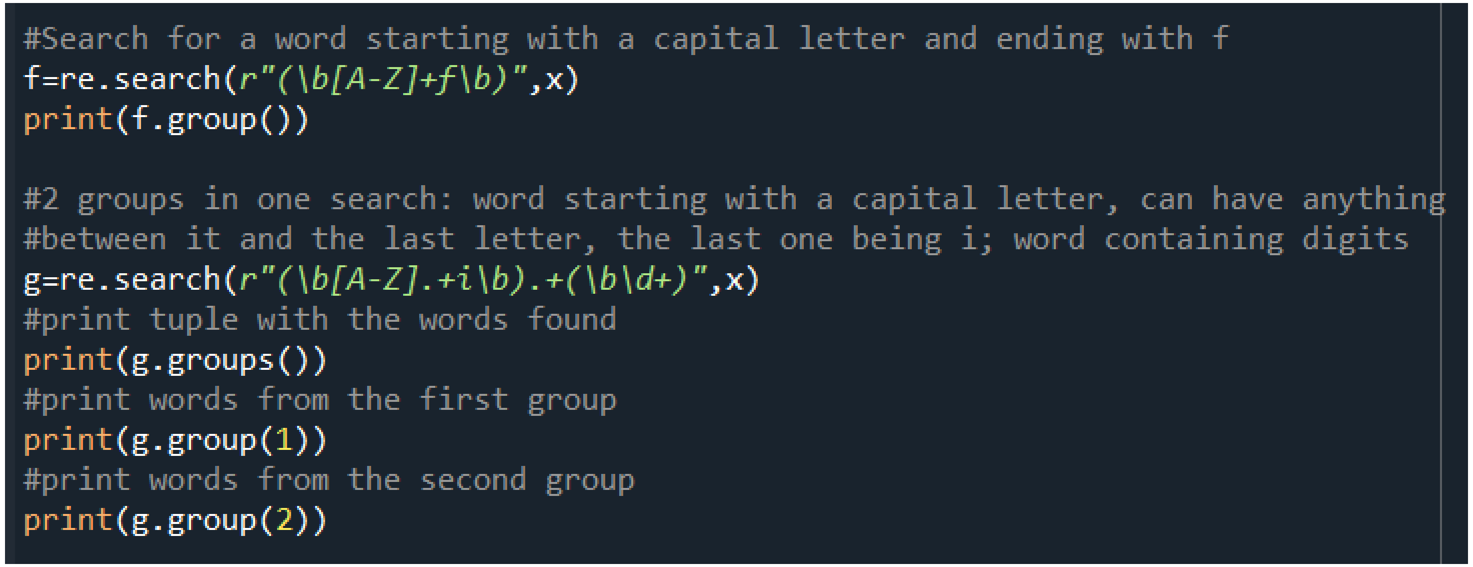


```
import re

x = "Mihai Eminescu-And If (1884)\
    And if the branches tap my pane \
    And the poplars whisper nightly,\
    It is to make me dream again \
    I hold you to me tightly. \
    And if the stars shine on the pond \
    And light its sombre shoal, \
    It is to quench my mind's despond \
    And flood with peace my soul. \
    And if the clouds their tresses part \
    And does the moon outblaze, \
    It is but to remind my heart \
    I long for you always."
```

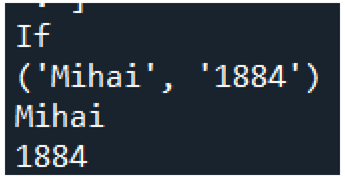


```
(0, 5)
Mihai Eminescu-And If (1884)    And if the branches tap my pane     And the
poplars whisper nightly,    It is to make me dream again     I hold you to me
tightly.     And if the stars shine on the pond     And light its sombre
shoal,     It is to quench my mind's despond     And flood with peace my soul.
And if the clouds their tresses part     And does the moon outblaze,     It is
but to remind my heart     I long for you always.
Mihai
['1', '4']
['Mi', 'If']
['q', 'z']
['M', ' ', 'E', 'u', '-', 'A', ' ', 'I', ' ', '(', '1', '8', '8', '4', ')', '
', ' ', ' ', ' ', 'A', ' ', ' ', 't', ' ', ' ', 't', ' ', 'y', ' ', ' ', ' ',
' ', ' ', ' ', 'A', ' ', 't', ' ', ' ', 'w', ' ', 't', 'y', ',', ' ', ' ', '
', ' ', 'I', 't', ' ', ' ', 't', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ', ' ',
'I', ' ', ' ', 'y', 'u', ' ', 't', ' ', ' ', 't', 't', 'y', '.', ' ', ' ', '
', ' ', ' ', 'A', ' ', ' ', 't', ' ', 't', ' ', ' ', ' ', 't', ' ', ' ', ' ',
' ', ' ', ' ', 'A', ' ', 't', ' ', 't', ' ', ' ', ',', ' ', ' ', ' ', ' ', '
', 'I', 't', ' ', ' ', 't', ' ', 'u', ' ', 'y', ' ', "'", ' ', ' ', ' ', ' ',
' ', ' ', 'A', ' ', ' ', 'w', 't', ' ', ' ', 'y', ' ', 'u', '.', ' ', ' ', '
', ' ', ' ', 'A', ' ', ' ', 't', ' ', 'u', ' ', 't', ' ', 't', ' ', 't', ' ',
' ', ' ', ' ', ' ', 'A', ' ', ' ', 't', ' ', ' ', 'u', 't', 'z', ',', ' ', '
', ' ', ' ', ' ', 'I', 't', ' ', ' ', 'u', 't', ' ', 't', ' ', ' ', 'y', ' ',
't', ' ', ' ', ' ', ' ', ' ', 'I', ' ', ' ', ' ', 'y', 'u', ' ', 'w', 'y',
'.']
```

Examples

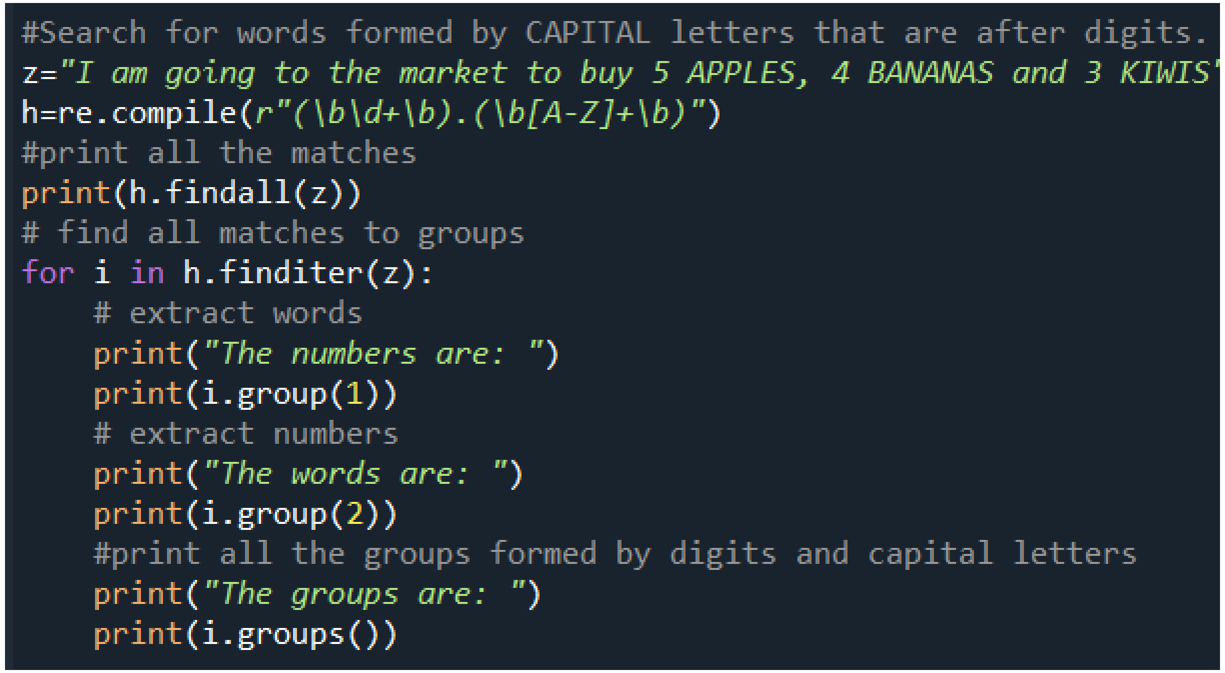


```
#Search for a word starting with a capital letter and ending with f
f=re.search(r"(\b[A-Z]+f\b)",x)
print(f.group())

#2 groups in one search: word starting with a capital letter, can have anything
#between it and the last letter, the last one being i; word containing digits
g=re.search(r"(\b[A-Z].+i\b).+(\b\d+)",x)
#print tuple with the words found
print(g.groups())
#print words from the first group
print(g.group(1))
#print words from the second group
print(g.group(2))
```

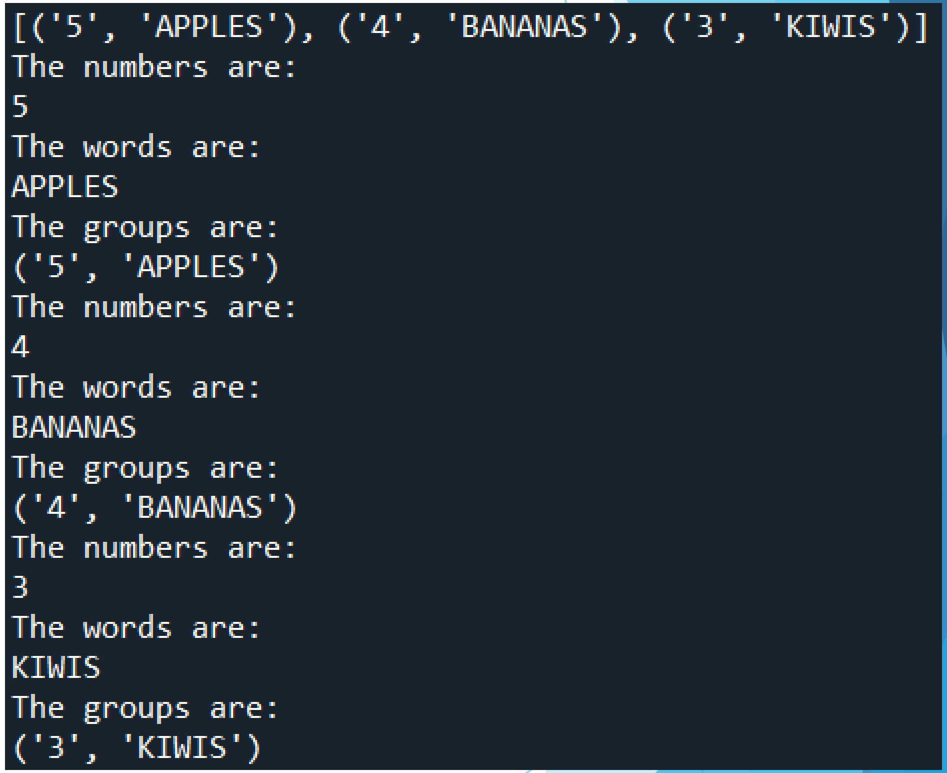


```
If
('Mihai', '1884')
Mihai
1884
```

Examples



```
#Search for words formed by CAPITAL letters that are after digits.
z="I am going to the market to buy 5 APPLES, 4 BANANAS and 3 KIWIS"
h=re.compile(r"(\b\d+\b).(\b[A-Z]+\b)")
#print all the matches
print(h.findall(z))
# find all matches to groups
for i in h.finditer(z):
    # extract words
    print("The numbers are: ")
    print(i.group(1))
    # extract numbers
    print("The words are: ")
    print(i.group(2))
    #print all the groups formed by digits and capital letters
    print("The groups are: ")
    print(i.groups())
```



```
[('5', 'APPLES'), ('4', 'BANANAS'), ('3', 'KIWIS')]
The numbers are:
5
The words are:
APPLES
The groups are:
('5', 'APPLES')
The numbers are:
4
The words are:
BANANAS
The groups are:
('4', 'BANANAS')
The numbers are:
3
The words are:
KIWIS
The groups are:
('3', 'KIWIS')
```

Exercises

1. a=“rectangle”, b=“square”, c=“sphere”, d=“triangle”, e=“cone”, f=“cube”, g=“cylinder”. Using compile(), print only the strings that start with either c or s and end with e.
2. words=“car, cat, dog, pool, bath, cone, cube, ring, int”. Write a regex that prints only the words that have exactly 4 letters.
3. list=[“square”,”triangle”,”cube”,”sphere”,”circle”,”pentagon”, “hexagon”, “rectangle”,”parallelogram”,”trapezoid”]. Loop thourgh the list and match only the words ending in “re”.
4. Take the list from the previous exercise and search for the words in it in the string: geo=“A square has 4 sides, a triangle has 3, a pentagon has 5, a hexagon has 6. While a square has 4 equal sides, a triangle can have 0, 2 or 3 equal sides.”. Extract from geo the digits and the non-digits characters.
5. link=“https://www.newyorker.com/magazine/2021/11/01/the-book-of-form-and-emptiness-the-war-for-gloria-read-until-you-understand-and-the-end-of-bias”. Extract the year, month and date from ex2.
6. date= "2021-11-02“. Change the date format to DD-MM-YYYY.
7. Write a function to check if a string starts with a digit. Provide at least two examples (one right and one wrong).
8. Write a function to check if a string ends with a digit. Provide at least two examples when calling the function (one right and one wrong).
9. Write a function to check if a string contains a digit. Provide at least two examples (one right and one wrong).

Homework

1. Write a function that checks if a string contains only lowercase letters, digits and *. Provide at least two examples (one right and one wrong).
2. Write a function that checks if a string has the following pattern: word containing only uppercase letters _ word containing only lowercase letters (e.g., FILS_student). Provide at least two examples (one right and one wrong).
3. hw4="rectangle square sphere triangle cone cube cylinder". Print all the words ending in "le" or "re".
4. Create a regex with at least two groups (like in the last example). Print all the matches in a string and all the matches in every group.
5. Create a program that changes the format of a date from YY-MM-DD to DD-MM-YY and changes the month from a number to its name. (e.g., 21-12-01 will be 01-December-21). Provide at least one example.
6. Write a function that matches only text that starts with m, ends with n and have exactly 3 characters between them. Provide at least two examples when calling the function (1 right and 1 wrong)
7. Write a function that matches only text that starts with h and is followed by exactly 2 or 3 i. Provide at least two examples when calling the function (1 right and 1 wrong)
8. Write a function that matches words containing q, but not in the beginning or at the end of a word. Provide at least two examples when calling the function (1 right and 1 wrong)
9. Write a function that replaces all a in a string with u and all i with e. (hint: you can use replace() or sub())