**REGULAR EXPRESSIONS EXERCISES SELF**

**PROGRAM 1**

print “PROGRAM checks for a sub-string ‘ads\_lift’ in 4 strings, and returns the string if found.”

stringslist = list() #list of 4 strings, only last one does not contain words ‘ads\_lift’

stringslist = ["ads\_lift\_holdout\_universal\_pictures\_fifty\_shades\_q1\_2017\_unvalidated enabled", "ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated enabled", "ads\_lift\_holdout\_universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated enabled", "universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated - enabled"]

import re

for i in range(len(stringslist)):

y = re.findall('^ads\_lift', stringslist[i])

if y > []:

print "The words, 'ads\_lift', was found in the string", stringslist[i]

**OUTPUT PROGRAM 1**

The words, 'ads\_lift', was found in the string ads\_lift\_holdout\_universal\_pictures\_fifty\_shades\_q1\_2017\_unvalidated enabled

The words, 'ads\_lift', was found in the string ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated enabled

The words, 'ads\_lift', was found in the string ads\_lift\_holdout\_universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated enabled

**PROGRAM 2: list of 4 strings, program reads back parts of strings *after* ‘ads\_lift’ substring**

stringslist = liststringslist = ["ads\_lift\_holdout\_universal\_pictures\_fifty\_shades\_q1\_2017\_unvalidated enabled", "ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated enabled", "ads\_lift\_holdout\_universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated enabled", "universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated - enabled"]

import re

for i in range(len(stringslist)):

y = re.findall('^ads\_lift', stringslist[i])

if y > []:

x = re.findall('ads\_lift([^$]\*)', stringslist[i])

print "The rest of the string after 'ads\_lift' in this line is:”, x[0]

**OUTPUT PROGRAM 2**

The rest of the string after 'ads\_lift' in this line is: \_holdout\_universal\_pictures\_fifty\_shades\_q1\_2017\_unvalidated enabled

The rest of the string after 'ads\_lift' in this line is: \_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated enabled

The rest of the string after 'ads\_lift' in this line is: \_holdout\_universal\_pictures\_dogs\_purpose\_q1\_2017\_unvalidated enabled

**PROGRAM 3:** Problem: find all vowels in string between consonants using findall

**See** <https://www.hackerrank.com/challenges/re-findall-re-finditer/problem> :

print "PROGRAM finds all vowels in string between consonants in a given string, 'rabcdeefgyYhFjkIoomnpOeorteeeeet'"

import re

stringI = "rabcdeefgyYhFjkIoomnpOeorteeeeet"

vowels = 'aeiouAEIOU'

match = list()

match = re.findall(r'['+vowels+']{2,}(?!\d)', stringI)

for i in range(len(match)):

print match[i]

**OUTPUT PROGRAM 3**

PROGRAM finds all vowels in string between consonants in a given string, 'rabcdeefgyYhFjkIoomnpOeorteeeeet'

ee

Ioo

Oeo

eeeee

**PROGRAM 4:** tests for 'X' in a list of 5 strings, if true, prints line and rest of the string

stringslist = ["X-Sieve: CMU Sieve 2.3 ", "AAAAAContent-Type-Message", "X-DSPAM-Result: Innocent ", "X-DSPAM-Confidence: 0.8475 ", "X-Content-Type-Message-Body: text/plain", ]

print "PROGRAM tests for 'X' in a list of 5 strings, if true, prints line and rest of the string."

import re

for i in range(len(stringslist)):

y = re.findall('^X', stringslist[i])

if y > []:

y = re.findall('**X[^$]\***', stringslist[i])

x = re.findall('**^X.\***', stringslist[i])

xstr = x[0]

xstr = xstr[1:] #Remove “X”, take rest of string

print "Line with 'X' in is:", stringslist[i], " and rest of the string after 'X' is:", xstr

**OUTPUT PROGRAM 4**

PROGRAM tests for 'X' in a list of 5 strings, if true, prints line and rest of the string.

Line with 'X' in is: X-Sieve: CMU Sieve 2.3 and rest of the string after 'X' is: -Sieve: CMU Sieve 2.3

Line with 'X' in is: X-DSPAM-Result: Innocent and rest of the string after 'X' is: -DSPAM-Result: Innocent

Line with 'X' in is: X-DSPAM-Confidence: 0.8475 and rest of the string after 'X' is: -DSPAM-Confidence: 0.8475

Line with 'X' in is: X-Content-Type-Message-Body: text/plain and rest of the string after 'X' is: -Content-Type-Message-Body: text/plain

**PROGRAM 5** : Find letters, punctuations marks, etc (whole string) and print them apart.

import re

line = '"ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated"ZZ: "XXenabledYY",'

y = re.findall('[^XYZ]',line)

if y <> []:

print y, "is extracted from line:", line

['"', 'a', 'd', 's', '\_', 'l', 'i', 'f', 't', '\_', 'h', 'o', 'l', 'd', 'o', 'u', 't', '\_', 'u', 'n', 'i', '\_', 'h', 'o', 'm', 'e', '\_', 'e', 'n', 't', '\_', 's', 'i', 'n', 'g', '\_', 'q', '1', '\_', '2', '0', '1', '7', '\_', 'u', 'n', 'v', 'a', 'l', 'i', 'd', 'a', 't', 'e', 'd', '"', ':', ' ', '"', 'e', 'n', 'a', 'b', 'l', 'e', 'd', '"', ','] is extracted from line: "ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated"ZZ: "XXenabledYY",

**Note, only lower case letters are find:**

import re

line = '"ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated"ZZ: "XXenabledYY",'

y = re.findall('[a-z0-9]',line)

if y <> []:

print y, "is extracted from line:", line

['a', 'd', 's', 'l', 'i', 'f', 't', 'h', 'o', 'l', 'd', 'o', 'u', 't', 'u', 'n', 'i', 'h', 'o', 'm', 'e', 'e', 'n', 't', 's', 'i', 'n', 'g', 'q', '1', '2', '0', '1', '7', 'u', 'n', 'v', 'a', 'l', 'i', 'd', 'a', 't', 'e', 'd', 'e', 'n', 'a', 'b', 'l', 'e', 'd'] is extracted from line: "ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidated"ZZ: "XXenabledYY",

**Now string has first been converted to lower case:**

import re

line = 'ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidatedZZ: XXenabledYY,'

linelower = line.lower()

y = re.findall('[a-z0-9]',linelower)

if y <> []:

print y, "is extracted from line:", line

['a', 'd', 's', 'l', 'i', 'f', 't', 'h', 'o', 'l', 'd', 'o', 'u', 't', 'u', 'n', 'i', 'h', 'o', 'm', 'e', 'e', 'n', 't', 's', 'i', 'n', 'g', 'q', '1', '2', '0', '1', '7', 'u', 'n', 'v', 'a', 'l', 'i', 'd', 'a', 't', 'e', 'd', 'z', 'z', 'x', 'x', 'e', 'n', 'a', 'b', 'l', 'e', 'd', 'y', 'y'] is extracted from lie: ads\_lift\_holdout\_uni\_home\_ent\_sing\_q1\_2017\_unvalidatedZZ: XXenabledYY,

**PROGRAM 6**

WS\_Egypt = list()

WS\_Egypt = ['River God', 'Desert God', 'Seventh', 'Warlock', 'The quest', 'Pharaoh']

match = list()

import re

for i in range(len(WS\_Egypt)):

word = re.findall(r'\w+ God',WS\_Egypt[i])

if len(word) > 0:

print word

for j in range(len(word)):

print word[j]

**OUTPUT PROGRAM 6**

['River God']

['Desert God']

**PROGRAM 7: Porgram tests if it is a valid telephone number.**

**(A valid mobile number is a ten digit number starting with a 7, 8 or 9.) The exercise has been altered:** <https://www.hackerrank.com/challenges/validating-the-phone-number?utm_campaign=challenge-recommendation&utm_medium=email&utm_source=24-hour-campaign>

import re

N=6

list\_no = ['95874562811', '1252478965', '8907896781', '6786867678', '7234123', 'XXXXYYYZZZ']

for i in range(N):

if len(list\_no[i]) == 10:

if re.match(r'[789]\d{9}$',list\_no[i]):

# \d Matches a Digit Character, {9}: digit can be matched 9 times, $ end of string

print 'YES,', list\_no[i], "is valid."

else:

print 'NO,', list\_no[i], "is not valid."

else: print 'NO,', list\_no[i], "is not valid, not correct length."

**OUTPUT PROGRAM 7**

NO, 95874562811 is not valid, not correct length.

NO, 1252478965 is not valid.

YES, 8907896781 is valid.

NO, 6786867678 is not valid.

NO, 7234123 is not valid, not correct length.

NO, XXXXYYYZZZ is not valid.