**Practice Quiz: Advanced Regular Expressions**

1.Question 1

We're working with a CSV file, which contains employee information. Each record has a name field, followed by a phone number field, and a role field. The phone number field contains U.S. phone numbers, and needs to be modified to the international format, with "+1-" in front of the phone number. Fill in the regular expression, using groups, to use the transform\_record function to do that.

import re

def transform\_record(record):

  new\_record = re.sub(\_\_\_)

  return new\_record

print(transform\_record("Sabrina Green,802-867-5309,System Administrator"))

# Sabrina Green,+1-802-867-5309,System Administrator

print(transform\_record("Eli Jones,684-3481127,IT specialist"))

# Eli Jones,+1-684-3481127,IT specialist

print(transform\_record("Melody Daniels,846-687-7436,Programmer"))

# Melody Daniels,+1-846-687-7436,Programmer

print(transform\_record("Charlie Rivera,698-746-3357,Web Developer"))

# Charlie Rivera,+1-698-746-3357,Web Developer

Solution:

2.Question 2

The multi\_vowel\_words function returns all words with 3 or more consecutive vowels (a, e, i, o, u). Fill in the regular expression to do that.

import re

def multi\_vowel\_words(text):

  pattern = \_\_\_

  result = re.findall(pattern, text)

  return result

print(multi\_vowel\_words("Life is beautiful"))

# ['beautiful']

print(multi\_vowel\_words("Obviously, the queen is courageous and gracious."))

# ['Obviously', 'queen', 'courageous', 'gracious']

print(multi\_vowel\_words("The rambunctious children had to sit quietly and await their delicious dinner."))

# ['rambunctious', 'quietly', 'delicious']

print(multi\_vowel\_words("The order of a data queue is First In First Out (FIFO)"))

# ['queue']

print(multi\_vowel\_words("Hello world!"))

# []

Solution:

3.Question 3

When capturing regex groups, what datatype does the groups method return?

1 point



A string



A tuple



A list



A float

4.Question 4

The transform\_comments function converts comments in a Python script into those usable by a C compiler. This means looking for text that begins with a hash mark (#) and replacing it with double slashes (//), which is the C single-line comment indicator. For the purpose of this exercise, we'll ignore the possibility of a hash mark embedded inside of a Python command, and assume that it's only used to indicate a comment. We also want to treat repetitive hash marks (##), (###), etc., as a single comment indicator, to be replaced with just (//) and not (#//) or (//#). Fill in the parameters of the substitution method to complete this function:

import re

def transform\_comments(line\_of\_code):

  result = re.sub(\_\_\_)

  return result

print(transform\_comments("### Start of program"))

# Should be "// Start of program"

print(transform\_comments("  number = 0   ## Initialize the variable"))

# Should be "  number = 0   // Initialize the variable"

print(transform\_comments("  number += 1   # Increment the variable"))

# Should be "  number += 1   // Increment the variable"

print(transform\_comments("  return(number)"))

# Should be "  return(number)"

Solution:

5.Question 5

The convert\_phone\_number function checks for a U.S. phone number format: XXX-XXX-XXXX (3 digits followed by a dash, 3 more digits followed by a dash, and 4 digits), and converts it to a more formal format that looks like this: (XXX) XXX-XXXX. Fill in the regular expression to complete this function.

import re

def convert\_phone\_number(phone):

  result = re.sub(\_\_\_)

  return result

print(convert\_phone\_number("My number is 212-345-9999.")) # My number is (212) 345-9999.

print(convert\_phone\_number("Please call 888-555-1234")) # Please call (888) 555-1234

print(convert\_phone\_number("123-123-12345")) # 123-123-12345

print(convert\_phone\_number("Phone number of Buckingham Palace is +44 303 123 7300")) # Phone number of Buckingham Palace is +44 303 123 7300

Solution: