1. What is the name of the feature responsible for generating Regex objects?

Ans: re.compile() function returns Regex objects

2. Why do raw strings often appear in Regex objects?

Ans: For not escaping the backslashes we use raw strings

3. What is the return value of the search() method?

Ans: It takes two parameters and returns a match object if there is a match. If there is more than one match, only the first occurrence of the match will be returned. If no matches are found, the value None is returned.

4. From a Match item, how do you get the actual strings that match the pattern?

Ans: Using group() method returns string of the matched text.

5. In the regex which created from the r'(\d\d\d)-(\d\d\d-\d\d\d\d)', what does group zero cover? Group 2? Group 1?

Ans: All the groups covers the elements in string which are digits. Group 0 is the entire match, group 2 covers the set of parentheses, and group 1 covers the first set of parentheses.

6. In standard expression syntax, parentheses and intervals have distinct meanings. How can you tell a regex that you want it to fit real parentheses and periods?

Ans: Using period character with re.findall()

7. The findall() method returns a string list or a list of string tuples. What causes it to return one of the two options?

Ans: A list of strings is returned when regex has no groups, and a list of tuples of strings is returned when regex has groups.

8. In standard expressions, what does the | character mean?

Ans: It means “or, either”

9. In regular expressions, what does the character stand for? (I didn’t get the question on which it is asking)

10.In regular expressions, what is the difference between the + and \* characters?

Ans: + means one or more matches, \* means zero or more matches.

11. What is the difference between {4} and {4,5} in regular expression?

Ans: {4} means four characters and {4,5} between 4 to 5 characters.

12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?

Ans: \d stands for matches any single digit, \w stands for matches any single letter, number or underscore and \s stands for matches a single whitespace character.

13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?

Ans: \D stands for matches any single character that is not a digit, \W stands for matches any single character that doesn’t match any single letter, number or underscore and \S stands for all non-white spaced symbols.

14. What is the difference between .\*? and .\*?

Ans: (.\*?) matches any character (.) any number of times (\*), as few times as possible to make the regex match (?). We’ll get match on any string, but you’ll only capture a blank string because of the question mark. (.\*)? Captures a group zero or one times (?). That group consists of a run of any length (\*) of any character(.). It will also match anything, but it’ll capture the first line, since the dot matches anything expect a newline.

15. What is the syntax for matching both numbers and lowercase letters with a character class?

Ans: we can use “[A-Za-z0-9]”

16. What is the procedure for making a normal expression in regax case insensitive?

Ans: We can use re.IGNORECASE

17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd argument in re.compile()?

Ans: . character matches any character except the newline character. If we pass re.DOTALL is passed as the second argument in re.compile(), then the dot will also match newline characters.

18. If numReg = re.compile(r'\d+'), what will numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') return?

Ans: It will return 'X drummers, X pipers, five rings, X hen'. Above lines replaces the digits with X

19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?

Ans: The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.

20. How would you write a regex that match a number with comma for every three digits? It must match the given following:

'42'

'1,234'

'6,368,745'

but not the following:

'12,34,567' (which has only two digits between the commas)

'1234' (which lacks commas)

Ans:

import re

sentence = input()

pattern = re.compile(r'\d{1,3}(,\d{3})\*')

matches = pattern.match(sentence)

if matches.group(0) != sentence:

print("No Match")

else:

print(matches.group(0) + " matches the pattern.")

21. How would you write a regex that matches the full name of someone whose last name is Watanabe? You can assume that the first name that comes before it will always be one word that begins with a capital letter. The regex must match the following:

'Haruto Watanabe'

'Alice Watanabe'

'RoboCop Watanabe'

but not the following:

'haruto Watanabe' (where the first name is not capitalized)

'Mr. Watanabe' (where the preceding word has a nonletter character)

'Watanabe' (which has no first name)

'Haruto watanabe' (where Watanabe is not capitalized)

Ans:

import re

sentence = input()

NameSearch = re.compile(r'\b[A-Z][a-zA-Z]+\sNakamoto\b', re.VERBOSE)

res = NameSearch.search(sentence)

print(res==None)

if res != None:

print(res.group())

22. How would you write a regex that matches a sentence where the first word is either Alice, Bob, or Carol; the second word is either eats, pets, or throws; the third word is apples, cats, or baseballs; and the sentence ends with a period? This regex should be case-insensitive. It must match the following:

'Alice eats apples.'

'Bob pets cats.'

'Carol throws baseballs.'

'Alice throws Apples.'

'BOB EATS CATS.'

but not the following:

'RoboCop eats apples.'

'ALICE THROWS FOOTBALLS.'

'Carol eats 7 cats.'

Ans:

import re

sentence = input()

strsearch = re.compile(r'(Alice|Bob|Carol)\s+(eats|pets|throws)\s+(apples|cats|baseballs)\.', re.IGNORECASE)

res = strsearch.search(sentence)

g = res.group()

print(g)