1. What is the name of the feature responsible for generating Regex objects?
2. Why do raw strings often appear in Regex objects?
3. What is the return value of the search() method?
4. From a Match item, how do you get the actual strings that match the pattern?
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?
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?
7. The findall() method returns a string list or a list of string tuples. What causes it to return one of
the two options?
8. In standard expressions, what does the | character mean?
9. In regular expressions, what does the character stand for?
10.In regular expressions, what is the difference between the + and * characters?
11. What is the difference between {4} and {4,5} in regular expression?
12. What do you mean by the \d, \w, and \s shorthand character classes signify in regular expressions?
13. What do means by \D, \W, and \S shorthand character classes signify in regular expressions?
14. What is the difference between .*? and .*?
15. What is the syntax for matching both numbers and lowercase letters with a character class?
16. What is the procedure for making a normal expression in regax case insensitive?
17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd
argument in re.compile()?
18. If numReg = re.compile(r'\d+'), what will numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4
hen') return?
19. What does passing re.VERBOSE as the 2nd argument to re.compile() allow to do?
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)
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)
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.'

1. The re.compile() function is responsible for generating Regex objects.
2. Raw strings are used in Regex objects because they allow backslashes to be interpreted as literal backslashes instead of escape characters. This is useful in regular expressions where backslashes are commonly used.
3. The search() method returns a Match object if the pattern is found in the searched string, otherwise it returns None.
4. The group() method of the Match object can be used to get the actual strings that match the pattern.
5. Group zero covers the entire matched string, while group 1 and group 2 cover the first and second groups of parentheses in the pattern, respectively.
6. To match real parentheses and periods in a regex, they can be escaped with a backslash. For example, to match a literal left parenthesis, use "(".
7. The findall() method returns a list of all non-overlapping matches in the searched string. If the pattern contains groups, it returns a list of tuples where each tuple contains the matched groups.
8. The | character in regular expressions means "or". It is used to specify multiple alternatives for a pattern to match.
9. The dot character (".") in regular expressions is a wildcard that matches any character except a newline.
10. The "+" character matches one or more occurrences of the preceding character or group, while the "*" character matches zero or more occurrences of the preceding character or group.
11. {4} matches exactly 4 occurrences of the preceding character or group, while {4,5} matches between 4 and 5 occurrences of the preceding character or group.

12. In regular expressions, \d represents any digit character (0-9), \w represents any word character (alphanumeric characters and underscore), and \s represents any whitespace character (spaces, tabs, newlines, etc.).
13. In regular expressions, \D represents any non-digit character, \W represents any non-word character, and \S represents any non-whitespace character.
14. "*.?" matches zero or more occurrences of any character (except newline) in a non-greedy way, while ".*" matches zero or more occurrences of any character (except newline) in a greedy way.
15. The syntax for matching both numbers and lowercase letters with a character class is "[a-z0-9]".
16. To make a normal expression case insensitive, the re.IGNORECASE flag can be passed as the second argument to re.compile().
17. The "." character in regular expressions normally matches any character except a newline. If re.DOTALL is passed as the second argument to re.compile(), it matches any character including a newline.
18. numRegex.sub('X', '11 drummers, 10 pipers, five rings, 4 hen') will return the string "X drummers, X pipers, five rings, X hen".
19. Passing re.VERBOSE as the second argument to re.compile() allows for the use of whitespace and comments within the regular expression, making it more readable and easier to understand.
20. The regex for matching numbers with commas every three digits is r'^\d{1,3}(,\d{3})*$'.
21. The regex for matching the full name of someone whose last name is Watanabe is r'^[A-Z][a-z]*\sWatanabe$'.
22. The regex for matching 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 is r'^(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs).$