Q1. What is the benefit of regular expressions?

Ans1- They allow for precise and flexible text search and manipulation. Some key benefits include:

Pattern Matching: Regular expressions enable you to define complex patterns to match text or data within a string, making it easier to find specific information.

Text Validation: They are useful for validating input data, such as email addresses, phone numbers, or dates, ensuring they meet specific criteria.

Text Extraction: You can extract specific parts of a text, like extracting email addresses from a document or extracting data from structured text.

Q2. Describe the difference between the effects of "(ab)c+" and "a(bc)+." Which of these, if any, is the unqualified pattern "abc+"?

Ans2- a) "(ab)c+": This pattern matches strings that start with "ab" and are followed by one or more occurrences of the letter "c." For example, it matches "abcc," "abccc," and so on.

b) "a(bc)+": This pattern matches strings that start with "a" followed by one or more occurrences of the sequence "bc." It matches "abc," "abcbc," "abcbcbc," and so on.

Q3. How much do you need to use the following sentence while using regular expressions?

import re

ans3- The line "import re" is necessary when using regular expressions in Python. It imports the "re" module, which provides functions and classes for working with regular expressions in Python. Without this import statement, you won't have access to the regular expression capabilities provided by the "re" module.

Q4. Which characters have special significance in square brackets when expressing a range, and under what circumstances?

Ans4- In square brackets within a regular expression, certain characters have special significance when expressing a range. The hyphen '-' is used to define a range of characters. For example, "[a-z]" represents all lowercase letters from 'a' to 'z'. To include a literal hyphen in the range, you can place it at the beginning or end of the brackets, like "[-a-z]" or "[a-z-]".

Q5. How does compiling a regular-expression object benefit you?

Ans5- Compiling a regular expression object in Python using the re.compile() function provides several benefits. It improves performance by pre-compiling the regular expression pattern into a bytecode representation, which can be reused for multiple matches, making subsequent searches faster.

Q6. What are some examples of how to use the match object returned by re.match and re.search?

Ans6- for re.match()

import re

pattern = re.compile(r'\d+')

result = pattern.match("12345")

print(result.group())

for re.search()

import re

pattern = re.compile(r'\d+')

result = pattern.search("The price is $123.45")

print(result.group())

Q7. What is the difference between using a vertical bar (|) as an alteration and using square brackets as a character set?

Ans7- The vertical bar | is used for alternation and matches one of the specified patterns. For example, the pattern a|b matches either "a" or "b". In contrast, square brackets [...] define a character set and match any single character from the set. For example, [abc] matches "a," "b," or "c". The main difference is that alternation is for matching entire patterns, while character sets are for matching individual characters.

Q8. In regular-expression search patterns, why is it necessary to use the raw-string indicator (r)? In   replacement strings?

Ans8- In replacement strings, using r is not necessary, as it primarily affects how the regular expression pattern is interpreted, not how the replacement string is processed. However, it's a good practice to use r for consistency and to avoid potential issues with escape sequences in the replacement string.