Q1. Explain the difference between greedy and non-greedy syntax with visual terms in as few words as possible. What is the bare minimum effort required to transform a greedy pattern into a non-greedy one? What characters or characters can you introduce or change?

Ans: In greedy approach regex pattern tend to consume maximum characters in a source string \* is a greedy quantifier. In non-greedy approach regex engine returns when it satisfies the matching criteria.To make a quantifier non-greedy append ?

Q2. When exactly does greedy versus non-greedy make a difference?  What if you're looking for a non-greedy match but the only one available is greedy?

Ans: So, the difference between the greedy and the non-greedy match is the following: The greedy match will try to match as many repetitions of the quantified pattern as possible. The non-greedy match will try to match as few repetitions of the quantified pattern as possible.

Q3. In a simple match of a string, which looks only for one match and does not do any replacement, is the use of a nontagged group likely to make any practical difference?

Q4. Describe a scenario in which using a nontagged category would have a significant impact on the program's outcomes.

Ans: an action word that identifies the performance to be demonstrated, a learning statement that specifies what learning will be demonstrated in the performance, a broad statement of the criterion or standard for acceptable performance.

Q5. Unlike a normal regex pattern, a look-ahead condition does not consume the characters it examines. Describe a situation in which this could make a difference in the results of your programme.

Ans: Lookahead is used as an assertion in Python regular expressions to determine success or failure whether the pattern is ahead i.e to the right of the parser's current position. They don't match anything. Hence, they are called as zero-width assertions.

Q6. In standard expressions, what is the difference between positive look-ahead and negative look-ahead?

Ans: The positive lookahead construct is a pair of parentheses, with the opening parenthesis followed by a question mark and an equal’s sign. negative lookahead in this type of lookahead the regex engine searches for a particular element which may be a character or characters or a group after the item matched.

Q7. What is the benefit of referring to groups by name rather than by number in a standard expression?

Ans: Named groups are handy because they let you use easily-remembered names, instead of having to remember numbers.

Q8. Can you identify repeated items within a target string using named groups, as in "The cow jumped over the moon"?

Ans: we can use findall() function to find the repeated items within a target string using named groups.

The above string contains “the” as repeated character

Q9. When parsing a string, what is at least one thing that the Scanner interface does for you that the re.findall feature does not?

Ans: re.findall() Return all non-overlapping matches of pattern in string, as a list of strings. The string is scanned left-to-right, and matches are returned in the order found.

Q10. Does a scanner object have to be named scanner?

Ans: yes we can name scanner object as scanner