**Assignment 7-**

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

**Ans-1** In Python feature responsible for generating regex objects is re.compile() function

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

**Ans-2** Raw strings are used so that backslashes do not have to be escaped.

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

**Ans-3** In Python the search() method returns Match objects

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

**Ans-**4 We use group() methods which will returns the part of the string where there was a match

**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-5** Group 0 is the entire match, group 1 covers the first set of parentheses, and group 2 covers the second 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-6** Periods and parentheses can be escaped with a backslash: \., \(, and \).

**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-7** When the regex has no groups, a list of strings is returned. When the regex has groups, a list of tuples of strings is returned.

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

**Ans-8** The | character signifies matching “either, or” between two groups.

**9. In regular expressions, what does the character stand for?**

**Ans-9**

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

**Ans-10** The difference between + and \* is The + matches one or more. The \* matches zero or more. Which means This operator repeats the smallest possible preceding regular expression as many times as necessary (including zero) to match the pattern.

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

**Ans-11** The {3} matches exactly three instances of the preceding group. The {3,5} matches between three and five instances.

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

**expressions?**

**Ans-12** The \d, \w, and \s shorthand character classes match a single digit, word, or space character, respectively.

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

**Ans-13** The \D, \W, and \S shorthand character classes match a single character that is not a digit, word, or space character, respectively.

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

**Ans-14** Passing re.I or re.IGNORECASE as the second argument to re.compile() will make the matching case insensitive.

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

**Ans-15** In the absence of re.DOT all, the . will match any character except newlines. In the case of re.DOT all, the dot will match newlines as well.

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

**Ans-**16 The .\* performs a greedy match, and the .\*? performs a nongreedy match.

**17. What does the . character normally match? What does it match if re.DOTALL is passed as 2nd**

**argument in re.compile()?**

**Ans-17** Either [0-9a-z] or [a-z0-9]

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

**hen) return?**

**Ans-18** it will return'X drummers, X pipers, five rings, X hens'

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

**Ans-19** The re.VERBOSE argument allows you to add whitespace and comments to the string passed to re.compile().

**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-20** re.compile(r'^\d{1,3}(,\d{3})\*$') will create this regex, but other regex strings can produce a similar regular expression.

**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-21** re.compile(r'[A-Z][a-z]\*\sNakamoto')

**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 22-** re.compile(r'(Alice|Bob|Carol)\s(eats|pets|throws)\s(apples|cats|baseballs)\.', re.IGNORECASE)