Programming in Python (CSE 3142)

MINOR ASSIGNMENT-6: STRINGS

1. Write a function that takes a string as a parameter and returns a string with every successive repetitive character replaced with a star(*). For example, ’balloon’ is returned as ’bal*o*n’.

2. Write a function that takes two strings and returns True if they are anagrams and False otherwise. A pair of strings is anagrams if the letters in one word can be arranged to form the second one.

3. Write a function that takes a sentence as an input parameter and displays the number of words in the sentence.

4. Write a function that takes a sentence as an input parameter and replaces the first letter of every word with the corresponding uppercase letter and rest of the letters in the word by corresponding letters in lowercase without using builtin function.

5. Write a function that takes a string as an input and determines the count of the number of words without using regular expression.

6. What will be the output on executing each of the statements, following the assignment statement:
   address = ’B-6, Lodhi road, Delhi’
   a. len(address)
   b. address[17:-1]
   c. address[-len(address): len(address)]
   d. address[:-12] + address[-12:]
   e. address.find(’delhi’)
   f. address.swapcase()
   g. address.split(’,’)
   h. address.isalpha()

7. Examine the following string:
   greeting = ’Good Morning. Have a Good Day!!’
   What will be the output for the following function calls:
   a. greeting.count(’Good’)
   b. greeting.find(’a’)
   c. greeting.rfind(’a’)
   d. greeting.capitalize()
   e. greeting.lower()
   f. greeting.upper()
   g. greeting.swapcase()
   h. greeting.istitle()
   i. greeting.replace(’Good’, ’Sweet’)
   j. greeting.strip()
   k. greeting.split()
   l. greeting.partition(’.’)

m. greeting.startswith(’good’)
n. greeting.endswith(’!!’)

8. Determine the patterns extracted by the following regular expressions.
   1. string1 = ’Python Programming Language’

   1. match1 = re.search(’......m?’, string1)
   print(match1.group())
   2. match2 = re.search(’......m{1,2}’, string1)
   print(match2.group())
   3. match3 = re.search(’.*Language$’, string1)
   print(match3.group())
   4. match4 = re.search(’\w*\s\w*’, string1)
   print(match4.group())
   5. match5 = re.search(’.*’, string1)
   print(match5.group())

   2. string2 = ’Car Number DL5645’

   1. match1 = re.search(’\w\w?\d{1,4}’, string2)
   print(match1.group())
   2. match2 = re.search(’.*5’, string2)
   print(match2.group())
   3. match3 = re.search(’.*5?’, string2)
   print(match3.group())
   4. match4 = re.search(’\d{3}’, string2)
   print(match4.group())
   5. match5 = re.search(’ˆ C.*5$’, string2)
   print(match5.group())

   3. string3 = ’cdcccdcddd343344aabb’
   1. match1 = re.search(’(c|d)*\d*(a|b)*’, string3)
   print(match1.group())
   2. match2 = re.search(’(cd)*d’, string3)
   print(match2.group())
   3. match3 = re.search(’(cc|cd)*(3|4)*(aa|bb)’, string3)
   print(match3.group())
   4. match4 = re.search(’(cc|cd|dd)*(3|4)*(aa|bb)’, string3)
   print(match4.group())
   5. match5 = re.search(’(cc|cd|dd)*(3|4)*(aa|bb)*’, string3)
   print(match5.group())

9. Write a python program to perform the following tasks:
   1. Reverse a string
   2. Reverse a string without reversing the words. Example:
   input data: ’welcome to iter’
   output: ’iter to welcome’

3. Check if a string is symmetric or asymmetric
4. Check if a string is palindrome.
5. Given a string s and index i, delete ith value from s
6. Count the number of vowels and consonants in a string.
7. Find length of a string without using inbuilt function.
8. Check if a string contains at least one digit and one alphabet.
9. Remove duplicates from a string.
10. Count frequency of characters in a string.
11. Find the character having maximum frequency in a string.
12. Check if a pair of strings are anagram to each other. Example: SILENT and LISTEN are anagrams.