

Python for Computer Science and Data Science 1 (CSE 3651)

MINOR ASSIGNMENT-7: 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 anagram 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. Create a program to count the number of occurrences of a specific character in a string.
5. Write a Python program to find the length of the longest word in a sentence.
6. Write a Python function that takes a string and returns a new string where every vowel in the input string is replaced by the next vowel in sequence (a → e, e → i, i → o, o → u, u → a).
7. Write a Python program that checks if a string is a "rotational palindrome." A rotational palindrome is a string that can be rearranged cyclically to form a palindrome.
8. Implement a program to check if a string is a valid URL.
9. Create a program to find the number of vowels and consonants in a string.
10. Write a script that reads a line of text as a string, tokenizes the string with the split method and outputs the tokens in reverse order. Use space characters as delimiters.
11. Write a script that reads a line of text, tokenizes the line using space characters as delimiters and outputs only those words beginning with the letter 'b' and ending with the letter 'd'.
12. Write a script that reads a five-letter word from the user and produces every possible three-letter string, based on the word's letters. For example, the "three-letter words" produced from the word "bathe" include "ate," "bat," "bet," "tab," "hat," "the," and "tea." Challenge: Investigate the functions from the itertools module, then use an appropriate function to automate this task.
13. Check whether a sentence contains more than one space between words. If so, remove the extra spaces and display the results. For example, 'Hello World' should become 'Hello World'.
14. Write a Python program to "reverse the middle half of characters in a string".
15. Write a Python program to print the substrings of a character having a particular frequency. For 'aabbccccddddd', you should print 'bbb' if particular frequency is 3.
16. Write a code to extract unique characters of a string in sorted order.
17. What are the outputs of the below codes and why?

(a) `s = "how now brown cow"`
`print(s[s.find('o'):s.rfind('o')]).`

Result: "ow now brown c".

(b) `chr(ord('A') + 2) + chr(ord('Z') - 3)` Concatenation: 'C' + 'W' = "CW".

(c) `s = "abc123def456ghi789"`

`indices = [i for i, c in enumerate(s) if c == '']`

`result = s[indices[1]+1:indices[2]] + s[indices[4]+1:]`

`print(result)`

The code has an error. The list comprehension tries to find empty characters (""), which don't exist in the string. Therefore, indices will be an empty list, and the code will fail.

(d) `s = "abracadabra"`

`print(s.replace(s[s.find('a'):s.find('r')], "XYZ"))` XYZracadXYZra

(e) `s = "hello"`

`shift = 2`

jgnnq

`print("".join(chr((ord(c) - 97 + shift) % 26 + 97) for c in s))`

(f) `s = "mississippi"`

`print("".join(sorted(set(s))))`

imps

18. What will be the output of executing each of the following statements after assigning the variable:

`quote = "The quick brown fox jumps over the lazy dog"`

(a) `quote.upper()`

(b) `quote[::-1]`

(c) `quote[4:19]`

(d) `quote.replace('fox', 'cat')`

(e) `quote.count('o')`

(f) `quote.startswith('The')`

(g) `'brown' in quote`

(h) `quote.islower()`

19. Examine the following string: `quote = 'Knowledge is power. Power is gained through knowledge.'` What will be the output for the following function calls:

(a) `quote.find('power')`

(b) `quote.rfind('knowledge')`

(c) `quote.title()`

(d) `quote.lower()`

(e) `quote.upper()`

(f) `quote.endswith('knowledge.')`

(g) `quote.split(' ')` ['Knowledge', 'is', 'power.', 'Power', 'is', 'gained', 'through', 'knowledge.']

(h) `quote.partition('is')` ('Knowledge', 'is', ' power. Power is gained through knowledge.')

(i) `quote.isalpha()`

False contains .

20. For `string1 = 'Python Programming Language'`, determine the patterns extracted by the following regular expressions:

(a) `match1 = re.search('. m?', string1)`

`print(match1.group())`

- (b) `match3 = re.search('.*Language$', string1)` "Python Programming Language"
`print(match3.group())`
- (c) `match4 = re.search(' w* s w*', string1)` No match (likely raises an `AttributeError` unless handled)
`print(match4.group())`
- (d) `match5 = re.search('.*', string1)` "Python Programming Language"
`print(match5.group())`

21. For `string1 = 'Python Programming Language'`, find the corresponding outputs.

- (a) `match1 = re.fullmatch(r'[A-Za-z]*', string1)`
`print(match1.group())` `AttributeError: 'NoneType' object has no attribute 'group'`
- (b) `match2 = re.sub(r'Programming', 'Coding', string1)`
`print(match2)` "Python Coding Language"
- (c) `match3 = re.split(r'\s+', string1)`
`print(match3)` ['Python', 'Programming', 'Language']
- (d) `match8 = re.findall(r'\w+', string1)`
`print(match8)` ['Python', 'Programming', 'Language']

22. Write a python program to check if a string is symmetric or asymmetric.

23. Given a string `s` and index `i`, write a python program to delete the `i`-th value from `s`.

24. Use regular expressions to validate secure passwords. Passwords must have a minimum of 8 characters and contain at least one each from uppercase characters, lowercase characters, digits, and punctuation characters, such as characters in `'!@#$$%&?*?'`.

25. Use regular expressions and the `findall` function to count the number of digits, non-digit characters, whitespace characters and words in a string.