International University
School of Computer Science and Engineering

Fundamentals of Programming

Lab 6
STRINGS

TASKS:

1. (Sorting Words in a Sentence) Write a script that reads a line of text as a string, tokenizes the string, and outputs the tokens in alphabetical order. Use space characters as delimiters.

2. (Tokenizing and Comparing Strings) Write a script that reads a line of text, tokenizes the line using space characters as delimiters and outputs only those words begin- ning with the letter 'b'.

3. (Tokenizing and Comparing Strings) Write a script that reads a line of text, tokenizes it using space characters as delimiters and outputs only those words ending with the letters 'ed'.

4. (Converting Integers to Characters) Use the c presentation type to display a table of the character codes in the range 0 to 255 and their corresponding characters.

5. (Converting Integers to Emojis) Modify the previous exercise to display 10 emojis beginning with the smiley face, which has the value 0x1F600:

   The value 0x1F600 is a hexadecimal (base 16) integer. See the online appendix “Number Systems” for information on the hexadecimal number system. You can find emoji codes by searching online for “Unicode full emoji list.” The Unicode website precedes each character code with "U+" (representing Unicode). Replace "U+" with "0x" to properly format the code as a Python hexadecimal integer.

6. (Counting Vowels in a Sentence) Write a script that reads a phrase and calculates the number of vowels used in it. To assess if a character in the string is a vowel, check if it is present in a second string. The second string consists of the vowels “aeiou.” For example, the letter “a” is present in the string “aeiou” and is thus considered a vowel. Count all the vowels in the given string and output this number to the user.

7. Write a script that enables the user to enter mathematical word problems like “two times three” and “seven minus five”, then use string processing to break apart the string into the numbers and the operation and return the result. So “two times three” would return 6 and “seven minus five” would return 2. To keep things simple, assume the user enters only the words for the numbers 0 through 9 and only the operations 'plus', 'minus', 'times' and 'divided by'.
8. (Regular Expressions: Replacing Substrings) Use regular expressions to change the name in a line from a poem.
   ‘Jennifer, you are more beautiful than the most beautiful rose’

9. Using Regex, write a script that reads a line of text, to- kenizes the line using space characters as delimiters and outputs only those words begin- ning with the letter 'b'.

10. Using Regex, write a script that reads a line of text, to- kenizes it using space characters as delimiters and outputs only those words ending with the letters 'ed'.

11. (Regular Expressions: Count Number of Occurrences of a Letter) Use the findall function to check the number of occurrences of the letter “a” in the sentence.
    ‘The mouse that the cat that the dog that the man frightened and chased ran away’

12. (Regular Expressions: Locating URLs) Use a regular expression to search through a string and to locate all valid URLs. For this exercise, assume that a valid URL has the form http://www.domain_name.extension, where extension must be two or more characters.

13. (Using Regular Expressions for Autocorrection) When we type a message on a smartphone, autocorrect changes the text we enter to remove errors. One common error is using two capital letters to start a capitalized word instead of one. Write a script that uses regular expressions to correct this error using the following algorithm:
    a) Search for all capitalized letters in the sample text.
    b) Check for each capitalized letter if the next letter is also capitalized.
    c) If this is the case, change the second letter to lowercase.
    d) Repeat this process until you reach the end of the phrase.
    Test your script with different phrases with different errors.

14. (Using Regular Expressions for Password Validation) When choosing a password, a user is often confronted with a set of rules they have to comply with in order to set a valid password. Write a script using the necessary regular expressions to check if a password is valid. For a chosen password to be valid, it should contain at least one lowercase and one uppercase letter and at least one number. The use of whites paces in a password is not al- lowed. Provide the necessary feedback to the user for their chosen password.