Міністерство освіти і науки України

Державний університет “Житомирська політехніка”

Кафедра інженерії програмного забезпечення

Група: ВТ-21-1[1]

Програмування мовою Python

Лабораторна робота № 3

«Розгалуження та цикли»

Виконав: Бабушко А. С.

Прийняв: Морозов Д. С.

***Мета роботи:*** ознайомитися зі рядками в мові Python, діями над ними.

***Хід роботи:***

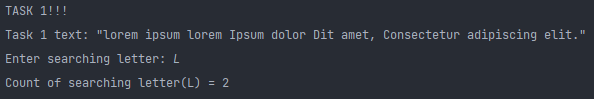
***Завдання на лабораторну роботу:***

******

***Лістинг програми:***

*""" Lab 3. Python. Andrii Babushko. Repository: https://github.com/AndriiBabushko/Python """  
  
  
# task 1  
def* task\_1\_find\_word\_in\_text(text, searching\_letter):  
 count\_searching\_letters = 0  
 searching\_letter = searching\_letter.capitalize()  
 split\_text = text.split(' ')  
 *for* i *in* range(0, len(split\_text)):  
 split\_text[i] = split\_text[i].capitalize()  
 *if* split\_text[i][0] == searching\_letter:  
 count\_searching\_letters += 1  
  
 *return* count\_searching\_letters  
  
  
print('\nTASK 1!!!')  
task\_1\_some\_text = 'lorem ipsum lorem Ipsum dolor Dit amet, Consectetur adipiscing elit.'  
print(f'Task 1 text: "{task\_1\_some\_text}"')  
task\_1\_some\_searching\_letter = str(input('Enter searching letter: '))  
print(f'Count of searching letter({task\_1\_some\_searching\_letter}) = '  
 f'{task\_1\_find\_word\_in\_text(task\_1\_some\_text, task\_1\_some\_searching\_letter)}')

***Результат програми:***

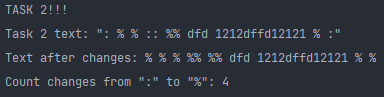
******

******

***Лістинг програми:***

*# task 2  
def* task\_2\_change\_text(text):  
 new\_text = text.replace(':', '%')  
 print(f'Text after changes: {new\_text}')  
 *return* text.count(':')  
  
  
print('\nTASK 2!!!')  
task\_2\_some\_text = ': % % :: %% dfd 1212dffd12121 % :'  
print(f'Task 2 text: "{task\_2\_some\_text}"')  
print(f'Count changes from ":" to "%": {task\_2\_change\_text(task\_2\_some\_text)}')

***Результат програми:***

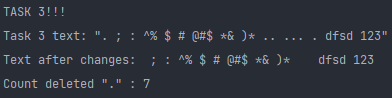
******

******

***Лістинг програми:***

*# task 3  
def* task\_3\_delete\_dots(text):  
 new\_text = text.replace('.', '')  
 print(f'Text after changes: {new\_text}')  
 *return* text.count('.')  
  
  
print('\nTASK 3!!!')  
task\_3\_some\_text = '. ; : ^% $ # @#$ \*& )\* .. ... . dfsd 123'  
print(f'Task 3 text: "{task\_3\_some\_text}"')  
print(f'Count deleted "." : {task\_3\_delete\_dots(task\_3\_some\_text)}')

***Результат програми:***

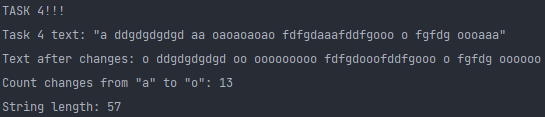
******

******

***Лістинг програми:***

*# task 4  
def* task\_4\_count\_a\_o(text):  
 new\_text = text.replace('a', 'o')  
 print(f'Text after changes: {new\_text}')  
 *return* [text.count('a'), len(new\_text)]  
  
  
print('\nTASK 4!!!')  
task\_4\_some\_text = 'a ddgdgdgdgd aa oaoaoaoao fdfgdaaafddfgooo o fgfdg oooaaa'  
print(f'Task 4 text: "{task\_4\_some\_text}"')  
task\_4\_result = task\_4\_count\_a\_o(task\_4\_some\_text)  
print(f'Count changes from "a" to "o": {task\_4\_result[0]}')  
print(f'String length: {task\_4\_result[1]}')

***Результат програми:***

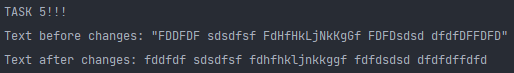
******

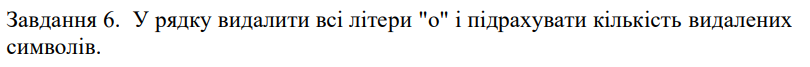
******

***Лістинг програми:***

*# task 5  
def* task\_5\_change\_upper\_lower(text):  
 new\_text = text.lower()  
 *return* new\_text  
  
  
print('\nTASK 5!!!')  
task\_5\_some\_text = 'FDDFDF sdsdfsf FdHfHkLjNkKgGf FDFDsdsd dfdfDFFDFD'  
print(f'Text before changes: "{task\_5\_some\_text}"')  
print(f'Text after changes: {task\_5\_change\_upper\_lower(task\_5\_some\_text)}')

***Результат програми:***

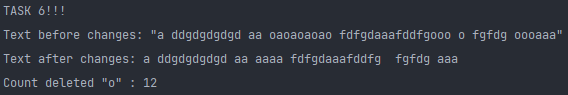
******

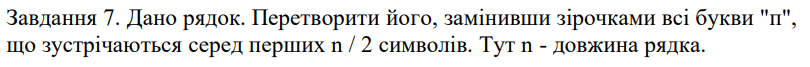
******

***Лістинг програми:***

*# task 6  
def* task\_6\_delete\_o(text):  
 new\_text = text.replace('o', '')  
 *return* [new\_text, text.count('o')]  
  
  
print('\nTASK 6!!!')  
task\_6\_some\_text = 'a ddgdgdgdgd aa oaoaoaoao fdfgdaaafddfgooo o fgfdg oooaaa'  
print(f'Text before changes: "{task\_6\_some\_text}"')  
task\_6\_result = task\_6\_delete\_o(task\_6\_some\_text)  
print(f'Text after changes: {task\_6\_result[0]}')  
print(f'Count deleted "o" : {task\_6\_result[1]}')

***Результат програми:***

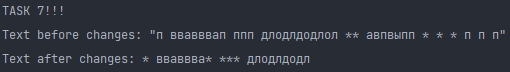
******

******

***Лістинг програми:***

*# task 7  
def* task\_7\_delete\_o(text):  
 half\_text = text[: int(len(text) / 2)]  
 new\_text = half\_text.replace('п', '\*')  
 *return* new\_text  
  
  
print('\nTASK 7!!!')  
task\_7\_some\_text = 'п ввавввап ппп длодлдодлол \*\* авпвыпп \* \* \* п п п'  
print(f'Text before changes: "{task\_7\_some\_text}"')  
task\_7\_result = task\_7\_delete\_o(task\_7\_some\_text)  
print(f'Text after changes: {task\_7\_result}')

***Результат програми:***

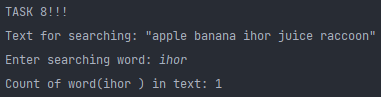
******

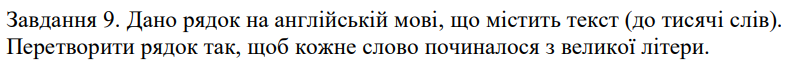
******

***Лістинг програми:***

*# task 8  
def* task\_8\_get\_word\_count\_in\_text(text, searched\_word):  
 *return* text.count(searched\_word)  
  
  
print('\nTASK 8!!!')  
task\_8\_some\_text = 'apple banana ihor juice raccoon'  
print(f'Text for searching: "{task\_8\_some\_text}"')  
task\_8\_searching\_word = input('Enter searching word: ')  
task\_8\_result = task\_8\_get\_word\_count\_in\_text(task\_8\_some\_text, task\_8\_searching\_word)  
print(f'Count of word({task\_8\_searching\_word}) in text: {task\_8\_result}')

***Результат програми:***

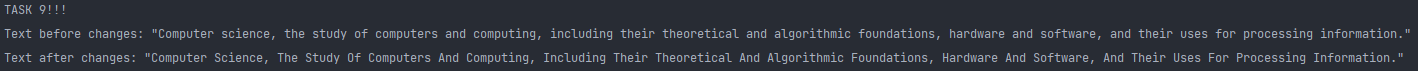
******

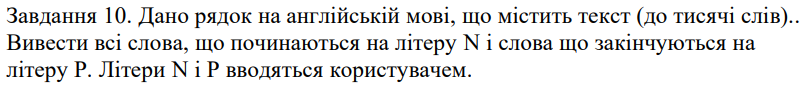
******

***Лістинг програми:***

*# task 9  
def* task\_9\_capitalize\_first\_word\_letters(text):  
 split\_text = text.split(' ')  
 *for* i *in* range(0, len(split\_text)):  
 split\_text[i] = split\_text[i].capitalize()  
 *return* ' '.join(split\_text)  
  
  
print('\nTASK 9!!!')  
task\_9\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text before changes: "{task\_9\_some\_text}"')  
print(f'Text after changes: "{task\_9\_capitalize\_first\_word\_letters(task\_9\_some\_text)}"')

***Результат програми:***

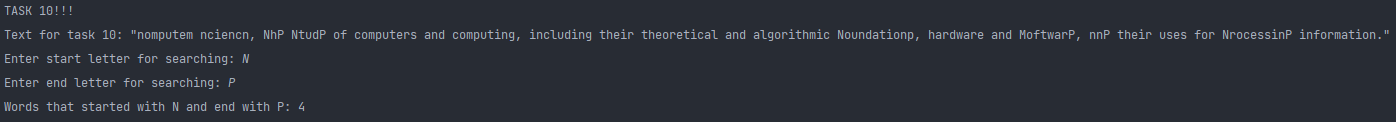
******

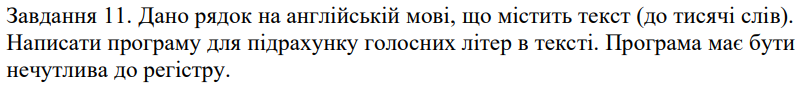
******

***Лістинг програми:***

*# task 10  
def* task\_10\_search\_words\_start\_end\_letters(text, start\_letter, end\_letter):  
 count\_words = 0  
 upper\_text = text.upper()  
 split\_text = upper\_text.split(' ')  
 start\_letter = start\_letter.upper()  
 end\_letter = end\_letter.upper()  
 *for* i *in* range(0, len(split\_text), 1):  
 *if* split\_text[i].startswith(start\_letter) *and* split\_text[i].endswith(end\_letter):  
 count\_words += 1  
 *else*:  
 *continue  
 return* count\_words  
  
  
print('\nTASK 10!!!')  
task\_10\_some\_text = 'nomputem nciencn, NhP NtudP of computers and computing, including their theoretical and ' \  
 'algorithmic Noundationp, hardware and MoftwarP, nnP their uses for NrocessinP information.'  
print(f'Text for task 10: "{task\_10\_some\_text}"')  
task\_10\_start\_letter = str(input('Enter start letter for searching: '))  
task\_10\_end\_letter = str(input('Enter end letter for searching: '))  
task\_10\_result = task\_10\_search\_words\_start\_end\_letters(task\_10\_some\_text, task\_10\_start\_letter, task\_10\_end\_letter)  
*if* task\_10\_result > 0:  
 print(f'Words that started with {task\_10\_start\_letter} and end with {task\_10\_end\_letter}: {task\_10\_result}')  
*else*:  
 print('No words found')

***Результат програми:***

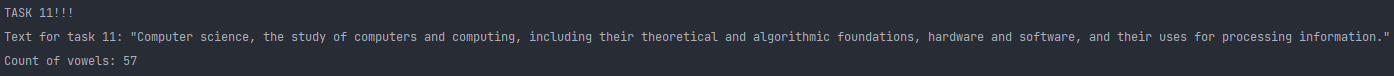
******

******

***Лістинг програми:***

*# task 11  
def* task\_11\_count\_vowels(text):  
 count\_vowels = 0  
 lower\_text = text.lower()  
 vowels = {'a', 'e', 'i', 'o', 'u'}  
 *for* string *in* lower\_text:  
 *if* string *in* vowels:  
 count\_vowels += 1  
  
 *return* count\_vowels  
  
  
print('\nTASK 11!!!')  
task\_11\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text for task 11: "{task\_11\_some\_text}"')  
  
task\_11\_result = task\_11\_count\_vowels(task\_11\_some\_text)  
*if* task\_11\_result > 0:  
 print(f'Count of vowels: {task\_11\_result}')  
*else*:  
 print('No vowels found')

***Результат програми:***

******

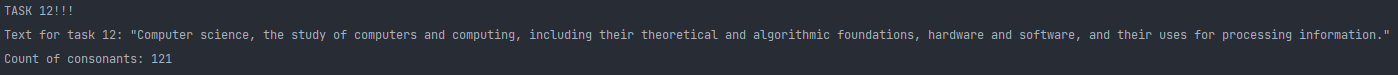
******

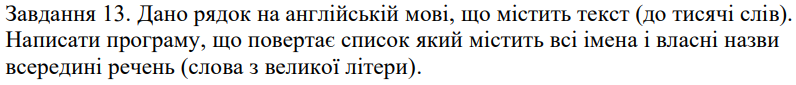
******

***Лістинг програми:***

*# task 12  
def* task\_12\_count\_consonant(text):  
 count\_consonant = 0  
 lower\_text = text.lower()  
 vowels = {'a', 'e', 'i', 'o', 'u'}  
 *for* string *in* lower\_text:  
 *if* string *in* vowels:  
 *continue  
 else*:  
 count\_consonant += 1  
  
 *return* count\_consonant  
  
  
print('\nTASK 12!!!')  
task\_12\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text for task 12: "{task\_12\_some\_text}"')  
  
task\_12\_result = task\_12\_count\_consonant(task\_12\_some\_text)  
*if* task\_12\_result > 0:  
 print(f'Count of consonants: {task\_12\_result}')  
*else*:  
 print('No consonants found')

***Результат програми:***

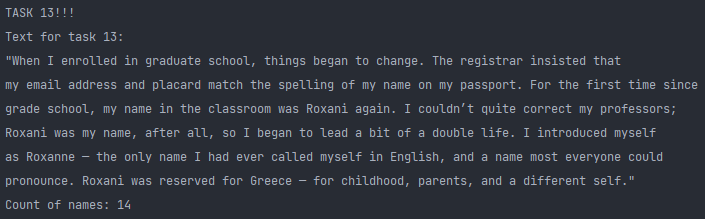
******

******

***Лістинг програми:***

*# task 13  
def* task\_13\_count\_names(text):  
 count\_names = 0  
 split\_text\_by\_dot = text.split('. ')  
 *for* sentence *in* split\_text\_by\_dot:  
 *for* index *in* range(0, len(sentence)):  
 *if* sentence[index].isupper():  
 count\_names += 1  
 *else*:  
 *continue  
  
 return* count\_names  
  
  
print('\nTASK 13!!!')  
task\_13\_some\_text = 'When I enrolled in graduate school, things began to change. The registrar insisted that \n' \  
 'my email address and placard match the spelling of my name on my passport. For the first time' \  
 ' since \ngrade school, my name in the classroom was Roxani again. I couldn’t quite correct ' \  
 'my professors; \nRoxani was my name, after all, so I began to lead a bit of a double life. ' \  
 'I introduced myself \nas Roxanne — the only name I had ever called myself in English, and' \  
 ' a name most everyone could \npronounce. Roxani was reserved for Greece — for childhood,' \  
 ' parents, and a different self.'  
print(f'Text for task 13:\n"{task\_13\_some\_text}"')  
  
task\_13\_result = task\_13\_count\_names(task\_13\_some\_text)  
*if* task\_13\_result > 0:  
 print(f'Count of names: {task\_13\_result}')  
*else*:  
 print('No names found')

***Результат програми:***

******

***Увесь лістинг програми:***

*""" Lab 3. Python. Andrii Babushko. Repository: https://github.com/AndriiBabushko/Python """  
  
  
# task 1  
def* task\_1\_find\_word\_in\_text(text, searching\_letter):  
 count\_searching\_letters = 0  
 searching\_letter = searching\_letter.capitalize()  
 split\_text = text.split(' ')  
 *for* i *in* range(0, len(split\_text)):  
 split\_text[i] = split\_text[i].capitalize()  
 *if* split\_text[i][0] == searching\_letter:  
 count\_searching\_letters += 1  
  
 *return* count\_searching\_letters  
  
  
print('\nTASK 1!!!')  
task\_1\_some\_text = 'lorem ipsum lorem Ipsum dolor Dit amet, Consectetur adipiscing elit.'  
print(f'Task 1 text: "{task\_1\_some\_text}"')  
task\_1\_some\_searching\_letter = str(input('Enter searching letter: '))  
print(f'Count of searching letter({task\_1\_some\_searching\_letter}) = '  
 f'{task\_1\_find\_word\_in\_text(task\_1\_some\_text, task\_1\_some\_searching\_letter)}')  
  
  
*# task 2  
def* task\_2\_change\_text(text):  
 new\_text = text.replace(':', '%')  
 print(f'Text after changes: {new\_text}')  
 *return* text.count(':')  
  
  
print('\nTASK 2!!!')  
task\_2\_some\_text = ': % % :: %% dfd 1212dffd12121 % :'  
print(f'Task 2 text: "{task\_2\_some\_text}"')  
print(f'Count changes from ":" to "%": {task\_2\_change\_text(task\_2\_some\_text)}')  
  
  
*# task 3  
def* task\_3\_delete\_dots(text):  
 new\_text = text.replace('.', '')  
 print(f'Text after changes: {new\_text}')  
 *return* text.count('.')  
  
  
print('\nTASK 3!!!')  
task\_3\_some\_text = '. ; : ^% $ # @#$ \*& )\* .. ... . dfsd 123'  
print(f'Task 3 text: "{task\_3\_some\_text}"')  
print(f'Count deleted "." : {task\_3\_delete\_dots(task\_3\_some\_text)}')  
  
  
*# task 4  
def* task\_4\_count\_a\_o(text):  
 new\_text = text.replace('a', 'o')  
 print(f'Text after changes: {new\_text}')  
 *return* [text.count('a'), len(new\_text)]  
  
  
print('\nTASK 4!!!')  
task\_4\_some\_text = 'a ddgdgdgdgd aa oaoaoaoao fdfgdaaafddfgooo o fgfdg oooaaa'  
print(f'Task 4 text: "{task\_4\_some\_text}"')  
task\_4\_result = task\_4\_count\_a\_o(task\_4\_some\_text)  
print(f'Count changes from "a" to "o": {task\_4\_result[0]}')  
print(f'String length: {task\_4\_result[1]}')  
  
  
*# task 5  
def* task\_5\_change\_upper\_lower(text):  
 new\_text = text.lower()  
 *return* new\_text  
  
  
print('\nTASK 5!!!')  
task\_5\_some\_text = 'FDDFDF sdsdfsf FdHfHkLjNkKgGf FDFDsdsd dfdfDFFDFD'  
print(f'Text before changes: "{task\_5\_some\_text}"')  
print(f'Text after changes: {task\_5\_change\_upper\_lower(task\_5\_some\_text)}')  
  
  
*# task 6  
def* task\_6\_delete\_o(text):  
 new\_text = text.replace('o', '')  
 *return* [new\_text, text.count('o')]  
  
  
print('\nTASK 6!!!')  
task\_6\_some\_text = 'a ddgdgdgdgd aa oaoaoaoao fdfgdaaafddfgooo o fgfdg oooaaa'  
print(f'Text before changes: "{task\_6\_some\_text}"')  
task\_6\_result = task\_6\_delete\_o(task\_6\_some\_text)  
print(f'Text after changes: {task\_6\_result[0]}')  
print(f'Count deleted "o" : {task\_6\_result[1]}')  
  
  
*# task 7  
def* task\_7\_delete\_o(text):  
 half\_text = text[: int(len(text) / 2)]  
 new\_text = half\_text.replace('п', '\*')  
 *return* new\_text  
  
  
print('\nTASK 7!!!')  
task\_7\_some\_text = 'п ввавввап ппп длодлдодлол \*\* авпвыпп \* \* \* п п п'  
print(f'Text before changes: "{task\_7\_some\_text}"')  
task\_7\_result = task\_7\_delete\_o(task\_7\_some\_text)  
print(f'Text after changes: {task\_7\_result}')  
  
  
*# task 8  
def* task\_8\_get\_word\_count\_in\_text(text, searched\_word):  
 *return* text.count(searched\_word)  
  
  
print('\nTASK 8!!!')  
task\_8\_some\_text = 'apple banana ihor juice raccoon'  
print(f'Text for searching: "{task\_8\_some\_text}"')  
task\_8\_searching\_word = input('Enter searching word: ')  
task\_8\_result = task\_8\_get\_word\_count\_in\_text(task\_8\_some\_text, task\_8\_searching\_word)  
print(f'Count of word({task\_8\_searching\_word}) in text: {task\_8\_result}')  
  
  
*# task 9  
def* task\_9\_capitalize\_first\_word\_letters(text):  
 split\_text = text.split(' ')  
 *for* i *in* range(0, len(split\_text)):  
 split\_text[i] = split\_text[i].capitalize()  
 *return* ' '.join(split\_text)  
  
  
print('\nTASK 9!!!')  
task\_9\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text before changes: "{task\_9\_some\_text}"')  
print(f'Text after changes: "{task\_9\_capitalize\_first\_word\_letters(task\_9\_some\_text)}"')  
  
  
*# task 10  
def* task\_10\_search\_words\_start\_end\_letters(text, start\_letter, end\_letter):  
 count\_words = 0  
 upper\_text = text.upper()  
 split\_text = upper\_text.split(' ')  
 start\_letter = start\_letter.upper()  
 end\_letter = end\_letter.upper()  
 *for* i *in* range(0, len(split\_text), 1):  
 *if* split\_text[i].startswith(start\_letter) *and* split\_text[i].endswith(end\_letter):  
 count\_words += 1  
 *else*:  
 *continue  
 return* count\_words  
  
  
print('\nTASK 10!!!')  
task\_10\_some\_text = 'nomputem nciencn, NhP NtudP of computers and computing, including their theoretical and ' \  
 'algorithmic Noundationp, hardware and MoftwarP, nnP their uses for NrocessinP information.'  
print(f'Text for task 10: "{task\_10\_some\_text}"')  
task\_10\_start\_letter = str(input('Enter start letter for searching: '))  
task\_10\_end\_letter = str(input('Enter end letter for searching: '))  
task\_10\_result = task\_10\_search\_words\_start\_end\_letters(task\_10\_some\_text, task\_10\_start\_letter, task\_10\_end\_letter)  
*if* task\_10\_result > 0:  
 print(f'Words that started with {task\_10\_start\_letter} and end with {task\_10\_end\_letter}: {task\_10\_result}')  
*else*:  
 print('No words found')  
  
  
*# task 11  
def* task\_11\_count\_vowels(text):  
 count\_vowels = 0  
 lower\_text = text.lower()  
 vowels = {'a', 'e', 'i', 'o', 'u'}  
 *for* string *in* lower\_text:  
 *if* string *in* vowels:  
 count\_vowels += 1  
  
 *return* count\_vowels  
  
  
print('\nTASK 11!!!')  
task\_11\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text for task 11: "{task\_11\_some\_text}"')  
  
task\_11\_result = task\_11\_count\_vowels(task\_11\_some\_text)  
*if* task\_11\_result > 0:  
 print(f'Count of vowels: {task\_11\_result}')  
*else*:  
 print('No vowels found')  
  
  
*# task 12  
def* task\_12\_count\_consonant(text):  
 count\_consonant = 0  
 lower\_text = text.lower()  
 vowels = {'a', 'e', 'i', 'o', 'u'}  
 *for* string *in* lower\_text:  
 *if* string *in* vowels:  
 *continue  
 else*:  
 count\_consonant += 1  
  
 *return* count\_consonant  
  
  
print('\nTASK 12!!!')  
task\_12\_some\_text = 'Computer science, the study of computers and computing, including their theoretical and ' \  
 'algorithmic foundations, hardware and software, and their uses for processing information.'  
print(f'Text for task 12: "{task\_12\_some\_text}"')  
  
task\_12\_result = task\_12\_count\_consonant(task\_12\_some\_text)  
*if* task\_12\_result > 0:  
 print(f'Count of consonants: {task\_12\_result}')  
*else*:  
 print('No consonants found')  
  
  
*# task 13  
def* task\_13\_count\_names(text):  
 count\_names = 0  
 split\_text\_by\_dot = text.split('. ')  
 *for* sentence *in* split\_text\_by\_dot:  
 *for* index *in* range(0, len(sentence)):  
 *if* sentence[index].isupper():  
 count\_names += 1  
 *else*:  
 *continue  
  
 return* count\_names  
  
  
print('\nTASK 13!!!')  
task\_13\_some\_text = 'When I enrolled in graduate school, things began to change. The registrar insisted that \n' \  
 'my email address and placard match the spelling of my name on my passport. For the first time' \  
 ' since \ngrade school, my name in the classroom was Roxani again. I couldn’t quite correct ' \  
 'my professors; \nRoxani was my name, after all, so I began to lead a bit of a double life. ' \  
 'I introduced myself \nas Roxanne — the only name I had ever called myself in English, and' \  
 ' a name most everyone could \npronounce. Roxani was reserved for Greece — for childhood,' \  
 ' parents, and a different self.'  
print(f'Text for task 13:\n"{task\_13\_some\_text}"')  
  
task\_13\_result = task\_13\_count\_names(task\_13\_some\_text)  
*if* task\_13\_result > 0:  
 print(f'Count of names: {task\_13\_result}')  
*else*:  
 print('No names found')

***Висновок:*** під час виконання лабораторної роботи було отримано навички роботи з рядками на мові Python з використанням допоміжних функцій та методів для роботи з цими рядками.