

Факультет: ПИиКТ

Информатика

\_Лабораторная работа\_№3

Вариант: разные, 3, 3

­­­­­

**Студент: Абрабоу Ахмед Елсаид А.И**

**№ группы: P3110**

**Номер студента: 333879**

**Преподаватель: Балакшин Павел Валерьевич**

**Санкт-Петербург**

**2021 г.**

**GitHub:**

**Задание:** (Смайлики)

A screenshot of a computer

Description automatically generated with medium confidence

Доп. задание №1 (+18 баллов)

**Вариант 3**

**Graphical user interface

Description automatically generated with low confidence**

**Вариант 1**

**Table

Description automatically generated**

**Доп. Задание №2.**

**Table

Description automatically generated**

**Исходный код**

import re  
  
s1 = [':', ';', 'X', '8', '=']  
s2 = ['-', '<', '-{', '<{']  
s3 = ['(', ')', 'O', '|', '\\', '/', 'P']  
isu = 333879  
print(s1[isu % 5] + s2[isu % 4] + s3[isu % 7])  
line='=<{('  
print(line.count('=<{('))  
line='<{s=<{(sdfsgsd=dsfg<{sdd'  
print(line.count('=<{('))  
line='<{(dfgghs{(zdf=<3{(cv=<{xv'  
print(line.count('=<{('))  
line='dfgghszdfcvxv=<{('  
print(line.count('=<{('))  
line='=<{(=<{(=<{(=<{(=<{(=<{('  
print(line.count('=<{('))  
print('\n')  
# the second part  
string = " Студент Вася вспомнил, что на своей лекции Балакшин П.В. упоминал про старшекурсников, которые будут ему помогать: Анищенко А.А. и Машина Е.А."  
regex = re.findall(r"([А-Я][а-я]+(?: [А-Я]+\.))", string)  
  
  
# print(regex)  
def listToString(regex):  
 # initialize an empty string  
 str1 = " "  
 # return string  
 return (str1.join(regex))  
  
  
# Driver code  
s = listToString(regex)  
regex1 = re.findall(r"([А-Я][а-я]+(?:))", s)  
  
  
def listToString(regex1):  
 # initialize an empty string  
 str1 = "\n"  
 # return string  
 return (str1.join(regex1))  
  
  
# Driver code  
print(listToString(regex1))  
print('\n')  
  
string = " Студент Вася вспомнил, что на своей лекции Балакшин П.В. Петров П.П. P000 Анищенко А.А. P33113 Примеров Е.В. P000 Иванов И.И. P000 упоминал про старшекурсников, которые будут ему помогать: Анищенко А.А. и Машина Е.А."  
regex = re.findall(r"([А-Я][а-я]+(?: [А-Я]+\.))", string)  
  
  
# print(regex)  
def listToString(regex):  
 # initialize an empty string  
 str1 = " "  
 # return string  
 return (str1.join(regex))  
  
  
# Driver code  
s = listToString(regex)  
regex1 = re.findall(r"([А-Я][а-я]+(?:))", s)  
  
  
def listToString(regex1):  
 # initialize an empty string  
 str1 = "\n"  
 # return string  
 return (str1.join(regex1))  
  
  
# Driver code  
print(listToString(regex1))  
print('\n')  
  
string = " В. Петров П.П. Анищенко А.А. Примеров Е.В. Иванов И.И. упоминал про старшекурсников, которые будут ему помогать: Анищенко А.А. и Машина Е.А."  
regex = re.findall(r"([А-Я][а-я]+(?: [А-Я]+\.))", string)  
  
  
# print(regex)  
def listToString(regex):  
 # initialize an empty string  
 str1 = " "  
 # return string  
 return (str1.join(regex))  
  
  
# Driver code  
s = listToString(regex)  
regex1 = re.findall(r"([А-Я][а-я]+(?:))", s)  
  
  
def listToString(regex1):  
 # initialize an empty string  
 str1 = "\n"  
 # return string  
 return (str1.join(regex1))  
  
  
# Driver code  
print(listToString(regex1))  
print('\n')  
string = "P3110"  
regex = re.findall(r"([А-Я][а-я]+(?: [А-Я]+\.))", string)  
  
  
# print(regex)  
def listToString(regex):  
 # initialize an empty string  
 str1 = " "  
 # return string  
 return (str1.join(regex))  
  
  
# Driver code  
s = listToString(regex)  
regex1 = re.findall(r"([А-Я][а-я]+(?:))", s)  
  
  
def listToString(regex1):  
 # initialize an empty string  
 str1 = "\n"  
 # return string  
 return (str1.join(regex1))  
  
  
# Driver code  
print(listToString(regex1))  
print('\n')  
  
  
string = "Дробыш Д.А. P3110"  
regex = re.findall(r"([А-Я][а-я]+(?: [А-Я]+\.))", string)  
  
  
# print(regex)  
def listToString(regex):  
 # initialize an empty string  
 str1 = " "  
 # return string  
 return (str1.join(regex))  
  
  
# Driver code  
s = listToString(regex)  
regex1 = re.findall(r"([А-Я][а-я]+(?:))", s)  
  
  
def listToString(regex1):  
 # initialize an empty string  
 str1 = "\n"  
 # return string  
 return (str1.join(regex1))  
  
  
# Driver code  
print(listToString(regex1))  
print('\n')  
  
  
  
from collections import Counter  
  
def remov\_duplicates(input):  
  
 # split input string separated by space  
 input = input.split(" ")  
  
 # joins two adjacent elements in iterable way  
 for i in range(0, len(input)):  
 input[i] = "".join(input[i])  
  
 # now create dictionary using counter method  
 # which will have strings as key and their  
 # frequencies as value  
 UniqW = Counter(input)  
  
 # joins two adjacent elements in iterable way  
 s = " ".join(UniqW.keys())  
 print (s)  
  
# Driver program  
if \_\_name\_\_ == "\_\_main\_\_":  
 input = 'Довольно распространённая ошибка ошибка – это лишний повтор повтор слова слова . Смешно, не не правда ли? Не нужно портить хор хоровод.'  
 remov\_duplicates(input)  
  
input = 'Довольно распространённая распространённая ошибка ошибка – это распространённая распространённая лишний повтор повтор слова слова . Смешно, не не правда ли? Не нужно портить хор хоровод.'  
remov\_duplicates(input)  
  
input = 'Довольно распространённая ошибка ошибка – это лишний повтор повтор правда слова слова . Смешно Смешно , не не правда правда ли? Не нужно портить хор хоровод.'  
remov\_duplicates(input)  
  
input = 'Довольно распространённая ошибка ошибка ошибка – это лишний лишний повтор повтор слова слова . Смешно, не не правда ли? Не нужно портить хор хоровод.'  
remov\_duplicates(input)  
  
input = 'Довольно распространённая ошибка ошибка ошибка – это лишний лишний повтор повтор слова слова . Смешно, не правда правда не правда ли? Не нужно портить хор хоровод.'  
remov\_duplicates(input)  
  
print("\n")  
  
# Task 2  
print("Task2")  
def Task2(line):  
 answer = []  
 fullNames = re.findall(r'[А-Я][а-я]{1,20}\s[A-Я]\.[А-Я]\.\sP[0-9]{1,20}', line)  
 a = re.findall(r'[A-Я]\.[А-Я]\.', line)  
 idx = 0  
 while len(a) > idx:  
 if a[idx][0] + a[idx][2] != a[idx][0] \* 2:  
 answer.append(fullNames[idx])  
 idx += 1  
 return answer  
  
  
def Tests\_For\_Task2():  
 flag = True  
 line = "Петров П.П. P000 Анищенко А.А. P33113 Примеров Е.В. P000 Иванов И.И. P000"  
 if (Task2(line)) == ["Примеров Е.В. P000"]:  
 print("Test 1 passed")  
 print(Task2(line))  
 else:  
 print("Test 1 Failed")  
 flag = False  
 line = "Афанасьева М.С. P3110 Бегинина А.А. P3110 Скворцова Д.А. P3110"  
 if (Task2(line)) == ['Афанасьева М.С. P3110', 'Скворцова Д.А. P3110']:  
 print("Test 2 passed")  
 print(Task2(line))  
 else:  
 print("Test 2 Failed")  
 flag = False  
 line = "Тюрин И.Н. P3110 Гребенник В.В. P3110 Пономарев И.М. P3110"  
 if (Task2(line)) == ['Тюрин И.Н. P3110', 'Пономарев И.М. P3110']:  
 print("Test 3 passed")  
 print(Task2(line))  
 else:  
 print("Test 3 Failed")  
 flag = False  
 line = "Дробыш Д.А. P3110"  
 if (Task2(line)) == ['Дробыш Д.А. P3110']:  
 print("Test 4 passed")  
 print(Task2(line))  
 else:  
 print("Test 4 Failed")  
 flag = False  
 print("'' -> None")  
 line = ""  
 if (Task2(line)) == []:  
 print("Test 5 passed")  
 print(Task2(line))  
 else:  
 print("Test 5 Failed")  
 flag = False  
  
  
Tests\_For\_Task2()

**Вывод0.1**

**=<{(**

**1**

**1**

**0**

**1**

**6**

**Вывод1.3**

**Балакшин**

**Анищенко**

**Машина**

**Балакшин**

**Петров**

**Анищенко**

**Примеров**

**Иванов**

**Анищенко**

**Машина**

**Петров**

**Анищенко**

**Примеров**

**Иванов**

**Анищенко**

**Машина**

**Дробыш**

**Вывод1.2**

**Довольно распространённая ошибка – это лишний повтор слова . Смешно, не правда ли? Не нужно портить хор хоровод.**

**Довольно распространённая ошибка – это лишний повтор слова . Смешно, не правда ли? Не нужно портить хор хоровод.**

**Довольно распространённая ошибка – это лишний повтор правда слова . Смешно , не ли? Не нужно портить хор хоровод.**

**Довольно распространённая ошибка – это лишний повтор слова . Смешно, не правда ли? Не нужно портить хор хоровод.**

**Довольно распространённая ошибка – это лишний повтор слова . Смешно, не правда ли? Не нужно портить хор хоровод.**

**Вывод2.3**

**Task2**

**Test 1 passed**

**['Примеров Е.В. P000']**

**Test 2 passed**

**['Афанасьева М.С. P3110', 'Скворцова Д.А. P3110']**

**Test 3 passed**

**['Тюрин И.Н. P3110', 'Пономарев И.М. P3110']**

**Test 4 passed**

**['Дробыш Д.А. P3110']**

**Test 5 passed**

**[]**