5.2 task 7

Дано:

Входные параметры:

Строка символов

Задание:



Выходные параметры:

Самая длинная последовательность подряд идущих букв “н” (включая заглавные)

Дополнительные условия:



Текст программы:

def main():

taskString = str()

# ввод строки

def str\_input():

taskString = str(input('Введите вашу строку\n'))

return taskString

# очистка строки

def str\_clean(taskString):

taskString = str()

return taskString

# подсчет последовательности "н"

def str\_count(taskString):

k = 0

s = 0

for i in range(0, len(taskString)): #нахождение символов и длины последовательности

if taskString[i] == 'н' or taskString[i] == 'Н':

k += 1

if k > s:

s = k

else:

k = 0

if s > 1: # проверка длины последовательности

print(f'Максимальная последовательность символа "н" в искомой строке: {s}\n')

elif s == 1:

print('В строке присутствуют только одиночные символы "н"\n')

elif s == 0:

print('Символов "н" в строке нет.\n')

# вывод строки из памяти

def str\_print(taskString):

print(taskString)

# пользовательский интерфейс

def menu():

ch = 1

while ch != 0:

ch = int(input('Выберите что вы хотите сделать:\n'

'1: Создать новую строку \n'

'2: Очистить строку \n'

'3: Подсчитать последовательность подряд идущих букв "н"\n'

'4: Вывод строки из памяти\n'

'0: Закрыть программу\n'))

if ch == 1:

taskString = str(str\_input())

elif ch == 2:

str\_clean(taskString)

elif ch == 3:

str\_count(taskString)

elif ch == 4:

str\_print(taskString)

menu()

main()

Результат тестирования:





5.2 task 7

Дано:

Входные параметры:

Строка символов

Задание:



Выходные параметры:

Подстрока, подходящая под вещественное число

Дополнительные условия:



Текст программы:

def main():

taskString = str()

# ввод строки

def str\_input():

taskString = str(input('Введите вашу строку\n'))

return taskString

# очистка строки

def str\_clean(taskString):

taskString = str()

return taskString

# выделение подстроки

def str\_find(taskString):

firstInd = int()

lastInd = int()

for i in range (0, len(taskString)):

try: # противодействие ошибке out of range

if taskString[i] == '.' and taskString[i+1].isdigit() and taskString[i-1].isdigit(): #проверка на наличие рядом цифр

j = i

try:

j = j + 1

while j <= len(taskString) and taskString[j].isdigit(): #нахождение правого конца подстроки

lastInd = j + 1

j += 1

except:

lastInd = j

try:

j = i - 1

while j >= 0 and taskString[j].isdigit(): # нахождение левого конца подстроки

firstInd = j

j -= 1

except:

firstInd = j

print(f'Искомая подстрока: {taskString[firstInd:lastInd]}')

except:

print('')

# вывод строки из памяти

def str\_print(taskString):

print(taskString)

# пользовательский интерфейс

def menu():

ch = 1

while ch != 0:

ch = int(input('Выберите что вы хотите сделать:\n'

'1: Создать новую строку \n'

'2: Очистить строку \n'

'3: Найти подстроку, соответствующую записи вещественного числа\n'

'4: Вывод строки из памяти\n'

'0: Закрыть программу\n'))

if ch == 1:

taskString = str(str\_input())

elif ch == 2:

str\_clean(taskString)

elif ch == 3:

str\_find(taskString)

elif ch == 4:

str\_print(taskString)

menu()

main()

Результат тестирования:

