

Тест по лекции 1

1 2 3 4

Вопрос 1

Что такое переменная?



Ваш ответ:

- ☐ Константа.
- ☒ Сущность для хранения данных.
- ☐ Источник случайных чисел.

Ответить

1 2 3 4

Вопрос 2

С помощью какой функции можно вывести данные в консоль?



Ваш ответ:

- ☒ print()
- ☐ printf()
- ☐ write()

Ответить

1 2 3 4

Вопрос 3

Можно ли в одну и ту же переменную класть данные разных типов, например, сначала записать строку, а потом число?



Ваш ответ:

- ☒ Да
- ☐ Нет

Ответить

1 2 3 4

Вопрос 4

С помощью какой функции можно считать данные пользователя?



Ваш ответ:

- ☐ scanf()
- ☒ input()
- ☐ insert()

Ответить

Тест по лекции 2

1 2 3 4 5

Вопрос 1



Выберите правильное определение переменной через тернарный оператор

Ваш ответ:

- ☐ a = b > c ? b : c
- ☒ a = b if b > c else c
- ☐ такого оператора не существует

Ответить

1 2 3 4 5

Вопрос 2



Каким образом в языке осуществляется выделение блоков кода?

Ваш ответ:

- ☐ С помощью фигурных скобок {}
- ☒ С помощью отступов
- ☐ С помощью ключевых слов begin и end

Ответить

1 2 3 4 5

Вопрос 3



С помощью какой встроенной функции можно генерировать последовательность чисел в заданном промежутке?

Ваш ответ:

- ☒ range
- ☐ numbers
- ☐ generate

Ответить

1 2 3 4 5

Вопрос 4



С помощью какого оператора реализуется проверка условия в коде?

Ваш ответ:

- ☐ while
- ☐ switch
- ☒ if

Ответить

1 2 3 4 5

Вопрос 5



С помощью какого оператора реализуется перебор значений в цикле?

Ваш ответ:

- ☒ for
- ☐ foreach
- ☐ cycle

Ответить

Тест по лекции 3

1

2

3

Вопрос 1



Какой из этих способов форматирования строк имеет место быть в Python (Переменная var = "World" определена ранее)?

Ваш ответ:

- ☐ "Hello %s" % var
- ☐ "Hello {}".format(var)
- ☐ f"Hello {var}!"
- ☒ Все способы подходят

Ответить

1

2

3

Вопрос 2



С помощью какого оператора можно добавить в свой код код из других библиотек и пользоваться им?

Ваш ответ:

- ☐ include
- ☐ addcode
- ☒ import

Ответить

1

2

3

Вопрос 3



С помощью какого оператора можно проверить, что подстрока содержится в строке?

Ваш ответ:

- ☐ contains
- ☒ in
- ☐ is_substr

Ответить

Тест по лекции 4

1 2 3

Вопрос 1

Какой принцип приведения типа списка к булеву типу?



Ваш ответ:

- ☐ Если все элементы истины, то и список истинен, иначе ложен.
- ☐ Список нельзя явно привести к булеву значению.
- ☒ Пустой список приводится ко лжи, в остальных случаях к истине.

Ответить

1 2 3

Вопрос 2

Можно ли в кортеж добавлять новые элементы?



Ваш ответ:

- ☐ Да
- ☒ Нет

Ответить

1 2 3

Вопрос 3

Какие из этих типов являются последовательностями?



Ваш ответ:

- ☐ int
- ☒ str
- ☒ list
- ☒ tuple

Ответить

Тест по лекции 5

1 2 3 4

Вопрос 1

Какими типами могут быть ключи у структуры словарь?



Ваш ответ:

- ☐ Только числа и строки.
- ☒ Только объекты, у которых есть метод hash()
- ☐ Любые объекты

Ответить

1 2 3 4

Вопрос 2

Какими типами могут быть значения у структуры словарь?



Ваш ответ:

- ☐ Только числа и строки.
- ☐ Только объекты, у которых есть метод hash()
- ☒ Любые объекты

Ответить

1 2 3 4

Вопрос 3

С помощью какого оператора можно пересечь две структуры множества?



Ваш ответ:

- ☐ |
- ☒ &
- ☐ ^

Ответить

1 2 3 4

Вопрос 4

Можно ли добавлять новые элементы к структуре словарь?



Ваш ответ:

- ☒ Да
- ☐ Нет

Ответить

Домашнее задание 1

Необходимо написать программу, которая будет считывать со стандартного ввода имя пользователя и выводить приветствие вида “Hello, [Username]”.

```
username = input()
print("Hello, {0}".format(username))
```

Домашнее задание 2

Необходимо написать программу, которая будет считывать со стандартного ввода название города и время и выводить информацию в формате “Current location is [Location] and time is [Time]”.

```
location = input()
time = input()
print("Current location is {0} and time is {1}".format(location, time))
```

Домашнее задание 3

Нужно написать программу, которая будет считывать со стандартного ввода целочисленные границы промежутка (сначала левая, потом правая, каждая на отдельной строке). А дальше будет считывать целые числа со стандартного ввода пока не встретит пустую строку, которая будет означать конец ввода. Нужно будет проверить входят ли все введенные числа в промежуток, проверка включает в себя границы промежутка.

```
a = int(input())
b = int(input())
i = input()
c = True
while i != "":
    if not(a <= int(i) <= b):
        c = False
    i = input()
print(c)
```

Домашнее задание 4

Необходимо будет написать программу, которая будет считывать со стандартного ввода строку и выводить на стандартный вывод является ли строка “правильной”. Строка считается правильной, если в ней есть латинская буква “a” или “o”, но нет букв “i” и “e”. Строка содержит только латинские буквы в нижнем регистре.

```
a = input()
if ("a" in a or "o" in a) and not("i" in a) and not("e" in a):
    print(True)
else:
    print(False)
```

Домашнее задание 5

Необходимо написать программу, которая будет считывать три числа и выводить их в определенном формате. Первое число целое, второе с плавающей точкой, третье целое неотрицательное. По примерам необходимо определить требуемый формат данных.

```
a = int(input())
b = float(input())
c = int(input())

print("{:+010d}".format(a))
print("{:#>10.2f}".format(b))
binary = "{:0>16}".format(bin(c)[2:])[::-1]
print("_".join(binary[i:i+4] for i in range(0, len(binary), 4))[::-1])
```

Домашнее задание 6

Необходимо написать программу, которая будет считывать со стандартного ввода строку, далее будет приводить ее к нижнему регистру, а также будет заменять символы “!”, “%”, “#”, “@” на пустую строку. В итоге нужно будет вывести в первой строке число замененных символов, а во второй преобразованную строку.

```
a = input()
count = a.count("!") + a.count("@") + a.count("#") + a.count("%")
a = a.lower()
a = a.replace("!", "").replace("@", "").replace("#", "").replace("%", "")
print(count)
print(a)
```

Домашнее задание 7

Необходимо написать программу, которая будет считывать со стандартного ввода строку. Нужно разбить строку на слова, словом будем считать последовательность символов отличных от пробелов (то есть знаки пунктуации будут входить в слова). Далее нужно посчитать и вывести среднее число символов в словах этого текста. Точность проверяется до 2го знака после запятой (точность ± 0.01).

* Решение можно реализовать в 2 строки кода.

```
a = input().split()
print(round(sum(map(len, a)) / len(a), 2))
```

Домашнее задание 8

Необходимо написать программу, которая будет считывать последовательности измерений. В каждой последовательности нужно выбрать максимальное значение, а в итоге вывести отсортированный по убыванию список этих макс значений, разделенных символом “;”. Во входных данных в первой строке будет задано целое положительное число – сколько записей нужно обработать, причем самих записей может быть больше чем это число, это нужно учесть. Значения в рамках одной записи разделены пробелом, минимальное число значений в записи – 1. Записи разделены переводом строки.

```
maxValues = []
count = int(input())
for i in range(0, count):
    values = list(map(int, input().split()))
    maxValues.append(sorted(values)[-1])

maxValues.sort(reverse=True)
result = ";".join(map(str, maxValues))
print(result)
```

Домашнее задание 9

Необходимо написать программу, которая будет считывать со стандартного ввода строку и выводить уникальные символы, встретившиеся в этой строке, в лексикографическом порядке по возрастанию. Символы из входа нужно приводить к нижнему регистру. Символ пробела не нужно учитывать при формировании результата. Выходные символы нужно разделять пробелом.

```
print(" ".join(sorted(set(input().replace(" ", "").lower()))))
```


Домашнее задание 10

Необходимо написать программу, которая будет считывать со стандартного ввода строку. Нужно разбить строку на слова, словом будем считать последовательность символов отличных от пробелов (то есть знаки пунктуации будут входить в слова). Далее нужно посчитать сколько каждое слово встречалось в тексте и вывести наиболее часто слово и сколько оно встретилось. Все слова нужно также приводить к нижнему регистру при подсчете. Гарантируется, что в тексте самое частое слово – единственное.

```
words = input().lower().split()
word_count = {}

for word in words:
    word_count[word] = word_count.get(word, 0) + 1

most_common_word = max(word_count, key=word_count.get)
count = word_count[most_common_word]

print(most_common_word, count)
```

Домашнее задание 11

Необходимо написать программу, которая будет принимать на вход строку, разбивать строку на слова по пробелу. Далее нужно из всех слов убрать следующие пунктуационные знаки:

!.,?;:#\$%^&*(),

а также привести слова к нижнему регистру. В итоге нужно вывести в алфавитном порядке слова, которые состоят как минимум из 5 символов, а также имеют как минимум 4 уникальных символа, и которые встретились в исходном тексте более 2х раз.

```
def remove_punctuation(input_string):
    punctuation_chars = "!.,?;:#$%^&*()"
    clean_string = ""
    for char in input_string:
        if char not in punctuation_chars:
            clean_string += char
    return clean_string

def process_string(input_string):
    input_string = remove_punctuation(input_string.lower())

    words = input_string.split()

    word_count = {}
    for word in words:
        if len(word) >= 5 and len(set(word)) >= 4:
            word_count[word] = word_count.get(word, 0) + 1

    filtered_words = [word for word, count in word_count.items() if count > 2]

    filtered_words.sort()
    return filtered_words

text = input().replace("\\", " ")
result = process_string(text)
for i in result:
    print(i)
```