**Лабораторная работа 1**

**Введение в язык программирования Python**

**Цель:** « Изучение среды разработки, типов данных, операторов и функций вывода и ввода данных на примере решения задач».

Содержание

[Полезные ссылки 1](#_Toc19204236)

[Задача 1 1](#_Toc19204237)

[Задача 2 2](#_Toc19204238)

[Задача 3 3](#_Toc19204239)

[Задача 4 4](#_Toc19204240)

[Задача 5 4](#_Toc19204241)

[Задача 6 6](#_Toc19204242)

[Задача 7 7](#_Toc19204243)

[Задача 8 8](#_Toc19204244)

[Задача 9 9](#_Toc19204245)

[Задача 10 9](#_Toc19204246)

[Задача 11 10](#_Toc19204247)

[Задача 12 11](#_Toc19204248)

[Требования к отчету по лабораторной работе 1: 11](#_Toc19204249)

# **Полезные ссылки:**

Официальный сайт Python, где можно скачать интерпретатор (Python 3): <https://www.python.org/>

Официальная документация по Python: <https://docs.python.org/3/>

Веб-сервис, позволяющий исполнять программы на Python прямо в вашем браузере: <https://trinket.io/python/41462f0f16>

Среда для написания программ PyCharm Educational Edition или PyCharm Community Edition:  
<https://www.jetbrains.com/pycharm-educational/>  
<https://www.jetbrains.com/pycharm/>

Текстовый редактор с подсветкой синтаксиса программ Sublime Text 3: <http://www.sublimetext.com/3>

Интерактивный учебник языка Python (на русском языке): <http://pythontutor.ru/>

Книга A Byte of Python: <http://www.swaroopch.com/notes/python/>  
и ее перевод на русский: <http://wombat.org.ua/AByteOfPython/>

Школа обучения Python (и не только) на английском:

<https://www.w3schools.com/python/python_try_except.asp>

Ещё один учебник онлайн: <https://www.geeksforgeeks.org/switch-case-in-python-replacement/>

# **Задача 1**

Вычислите в python значение выражения:

9\*\*19 - int(float(9\*\*19))

Обратить внимание, что используется одно и то же число, но результат получается не нулевой. Это показывает различие хранения большого числа в целочисленном типе и типе с плавающей точкой.

*Примечание:*

Если необходимо напечатать в консоль из файла, то тогда программа будет выглядеть следующим образом:

number = 9\*\*19 - int(float(9\*\*19))

print(number)

Пояснения по команде print() здесь: <https://www.w3schools.com/python/ref_func_print.asp>

Операция «\*\*» - возведение в степень числа, т.е.

# **Задача 2**

Напишите программу. Тимофей обычно спит ночью X часов и устраивает себе днем тихий час на Y минут. Определите, сколько всего минут Тимофей спит в сутки.

Внимание, программа принимает значения X и Y из стандартного потока ввода (функция input), результат надо выводить в стандартный поток вывода (функция print).

Для стандартного потока вывода используйте эту функцию без аргументов:

values = input() # Ввод строки с клавиатуры

*Примечание:*

Если необходимо ввести с клавиатуры числа, то функция ввода будет выглядеть следующим образом:

values = int(input()) # Ввод целого числа с клавиатуры

values = int(input(“Введите число”)) # Ввод целого числа с пояснением

**Пример входных данных (sample input):**

7

30

**Пример выходных данных (sample output):**

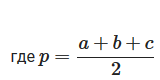
450

# **Задача 3**

В то далёкое время, когда Паша ходил в школу, ему очень не нравилась формула Герона для вычисления площади треугольника, так как казалась слишком сложной. В один прекрасный момент Павел решил избавить всех школьников от страданий и написать и распространить по школам программу, вычисляющую площадь треугольника по трём сторонам.

Одна проблема: так как эта формула не нравилась Павлу, он её не запомнил. Помогите ему завершить доброе дело и напишите программу, вычисляющую площадь треугольника по переданным длинам трёх его сторон по формуле Герона:

**

– полупериметр треугольника. На вход программе подаются целые числа, выводом программы должно являться вещественное число, соответствующее площади треугольника.

*Примечание:*

Для вычисления квадратного корня может быть использована функция math.sqrt, либо возведение в степень . Перед текстом программы, в случае использования метода из библиотеки, необходимо импортировать библиотеку math. Пример программы будет выглядеть следующим образом:

import math

a = int(input())

try:

    print(math.sqrt(a))

except ValueError as v:

    print("Корень из отрицательного числа не поддерживается")

Обработка исключений в языке Python подробнее:

<https://www.w3schools.com/python/python_try_except.asp>

**Пример входных данных (sample input):**

3

4

5

**Пример выходных данных (sample output):**

6.0

# **Задача 4**

Напишите программу, принимающую на вход целое число, которая выводит True, если переданное значение попадает в интервал (−15,12]∪(14,17)∪[19,+∞)

и False в противном случае (регистр символов имеет значение).

Обратите внимание на разные скобки, используемые для обозначения интервалов. В задании используются полуоткрытые и открытые интервалы. Подробнее про это вы можете прочитать, например, на википедии ([полуинтервал](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%BB%D1%83%D0%B8%D0%BD%D1%82%D0%B5%D1%80%D0%B2%D0%B0%D0%BB), [промежуток](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%BE%D0%BC%D0%B5%D0%B6%D1%83%D1%82%D0%BE%D0%BA_(%D0%BC%D0%B0%D1%82%D0%B5%D0%BC%D0%B0%D1%82%D0%B8%D0%BA%D0%B0)#.D0.A2.D0.B5.D1.80.D0.BC.D0.B8.D0.BD.D0.BE.D0.BB.D0.BE.D0.B3.D0.B8.D1.8F)).

*Примечание:*

Логический тип в Python записывается следующим образом: True, False. Для сравнения применяются *и* и *или*: and, or. Например:

number = int(input())

condition\_01 = number > -1 and number < 14

condition\_02 = number > 76

print(condition\_01 or condition\_02)

**Sample Input 1:**

20

**Sample Output 1:**

True

**Sample Input 2:**

-20

**Sample Output 2:**

False

# **Задача 5**

Напишите простой калькулятор, который считывает с пользовательского ввода три строки: первое число, второе число и операцию, после чего применяет операцию к введённым числам ("первое число" "операция" "второе число") и выводит результат на экран.

Поддерживаемые операции: +, -, /, \*, mod, pow, div, где   
mod — это взятие остатка от деления,   
pow — возведение в степень,   
div — целочисленное деление.

Если выполняется деление и второе число равно 0, необходимо выводить строку "Деление на 0!". Обратите внимание, что на вход программе приходят вещественные числа.

*Примечание:*

Операции в калькуляторе:

|  |  |
| --- | --- |
| Операция | Пояснение |
| + | Сложение |
| - | Вычитание |
| \* | Умножение |
| / | Деление |
| \*\* или pow | Возведение в степень |
| // или div | Целочисленное деление |
| % или mod | Взятие остатка от деления |

Один из вариантов решения задачи – вариант, где операции записываются в виде функций. Пример написания функции для деления:

def division (a, b):

    try:

        print(a / b)

    except ZeroDivisionError as ze:

        print("Деление на 0!")

Более простой вариант реализации – через условные операторы. Пример реализации условия if … else if … else … :

if operation == ‘+’:

print(a + b)

elif operation == ‘-’:

print(a - b)

else:

print(‘Операция не поддерживается в калькуляторе’)

**Реализация через switch-case операторы в Python отсутствует. Вместо них используются словари. Подробнее про словари тут** <https://www.w3schools.com/python/python_dictionaries.asp>

**Пример:**

def plus(a, b):

return a + b;

def minus(a, b):

return a - b;

operationDictionary = {

'+': plus, # операция сложения

'-': minus, # операция вычитания

}

a = float(input())

b = float(input())

operation = input()

calc = print(operationDictionary[operation](a, b))

**Sample Input 1:**

5.0

0.0

mod

**Sample Output 1:**

Деление на 0!

**Sample Input 2:**

-12.0

-8.0

\*

**Sample Output 2:**

96.0

**Sample Input 3:**

5.0

10.0

/

**Sample Output 3:**

0.5

# **Задача 6**

Напишите программу, которая получает на вход три целых числа, по одному числу в строке, и выводит на консоль в три строки сначала максимальное, потом минимальное, после чего оставшееся число.

На ввод могут подаваться и повторяющиеся числа.

*Примечание:*

Дополнительная информация по функции print(). Следующая строка

print(3, 1, 2, sep="\n")

Выведет числа каждое на следующей строке. Подробности про свойство sep тут

<https://www.w3schools.com/python/ref_func_print.asp>

**Sample Input 1:**

8

2

14

**Sample Output 1:**

14

2

8

**Sample Input 2:**

23

23

21

**Sample Output 2:**

23

21

23

# **Задача 7**

Паша очень любит кататься на общественном транспорте, а получая билет, сразу проверяет, счастливый ли ему попался. Билет считается счастливым, если сумма первых трех цифр совпадает с суммой последних трех цифр номера билета.  
  
Однако Паша очень плохо считает в уме, поэтому попросил вас написать программу, которая проверит равенство сумм и выведет "Счастливый", если суммы совпадают, и "Обычный", если суммы различны.  
  
На вход программе подаётся строка из шести цифр.

Выводить нужно только слово "Счастливый" или "Обычный", с большой буквы.

*Примечание:*

Условия в языке Python можно записывать как таким образом:

if something:

print(“Huraay!”)

else:

print(“Nooo!”)

Так и в обну строку:

print(“Huraay!” if something else “Nooo!”)

**Sample Input 1:**

090234

**Sample Output 1:**

Счастливый

**Sample Input 2:**

123456

**Sample Output 2:**

Обычный

# **Задача 8**

В институте биоинформатики по офису передвигается робот. Недавно студенты из группы программистов написали для него программу, по которой робот, когда заходит в комнату, считает количество программистов в ней и произносит его вслух: "n программистов".

Для того, чтобы это звучало правильно, для каждого *n*

нужно использовать верное окончание слова.

Напишите программу, считывающую с пользовательского ввода целое число *n*

(неотрицательное), выводящее это число в консоль вместе с правильным образом изменённым словом "программист", для того, чтобы робот мог нормально общаться с людьми, например: 1 программист, 2 программиста, 5 программистов.

В комнате может быть очень много программистов. Проверьте, что ваша программа правильно обработает все случаи, как минимум до 1000 человек.

**Sample Input 1:**

5

**Sample Output 1:**

5 программистов

**Sample Input 2:**

0

**Sample Output 2:**

0 программистов

**Sample Input 3:**

1

**Sample Output 3:**

1 программист

**Sample Input 4:**

2

**Sample Output 4:**

2 программиста

# **Задача 9**

Напишите программу, которая считывает с консоли числа (по одному в строке) до тех пор, пока сумма введённых чисел не будет равна 0 и **сразу после этого** выводит сумму квадратов всех считанных чисел.

Гарантируется, что в какой-то момент сумма введённых чисел окажется равной 0, **после этого считывание продолжать не нужно**.

В примере мы считываем числа 1, -3, 5, -6, -10, 13; в этот момент замечаем, что сумма этих чисел равна нулю и выводим сумму их квадратов, не обращая внимания на то, что остались ещё не прочитанные значения.﻿

*Примечание:*

Обратить внимание на цикл while. Подробнее здесь:

<https://www.w3schools.com/python/python_while_loops.asp>

**Sample Input:**

1

-3

5

-6

-10

13

4

-8

**Sample Output:**

340

# **Задача 10**

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... (число повторяется столько раз, чему равно). На вход программе передаётся неотрицательное целое число n — столько элементов последовательности должна отобразить программа. На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.

*Примечание:*

Обратить внимание на цикл for i in range(n, m). Подробнее здесь:

<https://www.w3schools.com/python/python_for_loops.asp>

**Sample Input:**

7

**Sample Output:**

1 2 2 3 3 3 4

# **Задача 11**

Напишите программу, которая считывает список чисел *lst* из первой строки и число *x* из второй строки, которая выводит все позиции, на которых встречается число *x* в переданном списке *lst*.

Позиции нумеруются с нуля, если число *x* не встречается в списке, вывести строку "Отсутствует" (без кавычек, с большой буквы).

Позиции должны быть выведены в одну строку, по возрастанию абсолютного значения.

*Примечание:*

Чтобы считать массив с клавиатуры можно воспользоваться следующей записью цикла for:

massive = [int(i) for i in input().split()]

**Sample Input 1:**

5 8 2 7 8 8 2 4

8

**Sample Output 1:**

1 4 5

**Sample Input 2:**

5 8 2 7 8 8 2 4

10

**Sample Output 2:**

Отсутствует

# **Задача 12**

Напишите программу, на вход которой подаётся прямоугольная матрица в виде последовательности строк, заканчивающихся строкой, содержащей только строку "end" (без кавычек)

Программа должна вывести матрицу того же размера, у которой каждый элемент в позиции i, j равен сумме элементов первой матрицы на позициях (i-1, j), (i+1, j), (i, j-1), (i, j+1). У крайних символов соседний элемент находится с противоположной стороны матрицы.

В случае одной строки/столбца элемент сам себе является соседом по соответствующему направлению.

*Примечание:*

Информация про двумерные массивы:

<https://snakify.org/en/lessons/two_dimensional_lists_arrays/>

**Sample Input 1:**

9 5 3

0 7 -1

-5 2 9

end

**Sample Output 1:**

3 21 22

10 6 19

20 16 -1

**Sample Input 2:**

1

end

**Sample Output 2:**

4

# **Требования к отчету по лабораторной работе 1:**

1. Титульный лист;
2. Для каждой задачи:
   1. формулировка;
   2. решение (алгоритм) для сложных задач;
   3. текст программы;
   4. скриншоты выполнения программы;
   5. вывод по решению задачи.

***Обращать внимание на примеры входных и выходных данных к задаче (sample input and sample output).***