**13.6 Практическая работа**

Цель домашнего задания

Научиться:

* Возвращать результат из функции - return;
* Передавать значения, объекты в функции, через аргументы;
* Изучить о способах возврата значений из функции – return
* Узнать о передаче объектов в функцию и поведение их, при изменении внутри функции.
* Использовать нескольких операторов return в функции.
* Научиться решать задачи с числами с плавающей точкой в экспоненциальном виде;
* Использовать точность – машинный epsilon, предельно малое различимое вещественное число, для решения задач.
* Использовать алгоритмы с заданной точностью для расчета;
* Узнать об особенностях сравнения чисел с плавающей точкой.

Что входит в задание

Задача 1. Урок информатики 2

Задача 2. Функция максимума

Задача 3. Число наоборот 2

Задача 4. Урок информатики 3

Задача 5. Недоделка 2

Задача 6. Маятник

Задача 7. Яйца

Задача 8. Сумма ряда

Задача 9. Аннуитетный платёж

Задача 1. Урок информатики 2

**Что нужно сделать**

В прошлый раз учитель написал программу, которая выводит числа в формате плавающей точки, однако он вспомнил, что не учёл одну важную штуку: числа-то могут идти от нуля.

Задано положительное число x (x > 0). Ваша задача преобразовать его в формат плавающей точки, то есть x = a · 10^b, где 1 ≤ a < 10. Обратите внимание, что x теперь больше нуля, а не больше единицы. Обеспечьте контроль ввода.

**Пример 1:**

Введите число: 92345

Формат плавающей точки: x = 9.2345 \* 10 \*\* 4

**Пример 2:**

Введите число: 0.0012

Формат плавающей точки: x = 1.2 \* 10 \*\* -3

**Что оценивается**

* результат вывода соответствует условию;
* обеспечен контроль ввода и обрабатываемые числа > 0;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 2. Функция максимума

**Что нужно сделать**

Юра пишет различные полезные функции для Питона, чтобы остальным программистам стало проще работать. Он захотел написать функцию, которая будет находить максимум из перечисленных чисел. Функция для нахождения максимума из двух чисел у него уже есть. Юра задумался: может быть, её можно как-то использовать для нахождения максимума уже от трёх чисел?

Помогите написать Юре, которая находит максимум из трёх чисел. Для этого используйте только функцию нахождения максимума из двух чисел

**Что оценивается**

* результат вывода соответствует условию;
* Найден корректный максимум трех чисел;
* input содержит корректное приглашение для ввода;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 3. Число наоборот 2

**Что нужно сделать**

Пользователь вводит два числа — N и K. Напишите программу, которая заменяет каждое число на число, которое получается из исходного записью его цифр в обратном порядке, затем складывает их, снова переворачивает и выводит ответ на экран.

**Пример:**

Введите первое число: 102

Введите второе число: 123

Первое число наоборот: 201

Второе число наоборот: 321

Сумма: 522

Сумма наоборот: 225

**Что оценивается**

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 4. Урок информатики 3

**Что нужно сделать**

Наконец-то учитель смог объяснить детям, что такое эта «плавающая точка». Однако долго его радость не продлилась, ведь на следующем уроке нужно будет объяснить такие страшные слова, как «экспоненциальное», «мантисса» и «порядок». Хоть старшеклассники и знакомы с экспонентой, учитель всё равно не уверен, что здесь всё будет понятно. Поэтому для наглядности он также написал программу.

На вход подаётся строка — это экспоненциальная форма числа. Напишите программу, которая выводит отдельно мантиссу и отдельно порядок этого числа.

**Что оценивается**

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 5. Недоделка 2

**Что нужно сделать**

Вы всё так же работаете в конторе по разработке игр и смотрите различные программы прошлого горе-программиста. В одной из игр для детей, связанной с мультяшной работой с числами, вам нужно было написать код по следующим условиям: программа получает на вход два числа. В первом числе должно быть не меньше трёх цифр, во втором числе — не меньше четырёх, иначе программа выдаёт ошибку. Если всё нормально, то в каждом числе первая и последняя цифра меняются местами, а затем выводится их сумма.

И тут вы натыкаетесь на программу, которая была написана прошлым программистом и которая как раз решает такую задачу! Однако старший программист сказал вам немного переписать этот код, чтобы он не выглядел так ужасно. Да и вам самим становится, мягко говоря, не по себе от него.

Разбейте приведённую ниже программу на функции. Повторений кода должно быть как можно меньше. Также сделайте, чтобы в основной части программы был только ввод чисел, затем изменённые числа и вывод их суммы.

first\_n = int(input("Введите первое число: "))  
first\_num\_count = 0  
temp = first\_n  
while temp > 0:  
    first\_num\_count += 1  
    temp = temp // 10     
if first\_num\_count < 3:  
 print("В первом числе меньше трёх цифр.")  
else:  
 last\_digit = first\_n % 10  
 first\_digit = first\_n // 10 \*\* (first\_num\_count - 1)  
 between\_digits = first\_n % 10 \*\* (first\_num\_count - 1) // 10  
 first\_n = last\_digit \* 10 \*\* (first\_num\_count - 1) + between\_digits \* 10 + first\_digit  
  
 print('Изменённое первое число:', first\_n)  
  
 second\_n = int(input("\nВведите второе число: "))  
  
 second\_num\_count = 0  
 temp = second\_n  
 while temp > 0:  
   second\_num\_count += 1  
   temp = temp // 10     
  
 if second\_num\_count < 4:  
   print("Во втором числе меньше четырёх цифр.")  
 else:  
   last\_digit = second\_n % 10  
   first\_digit = second\_n // 10 \*\* (second\_num\_count - 1)  
   between\_digits = second\_n % 10 \*\* (second\_num\_count - 1) // 10  
   second\_n = last\_digit \* 10 \*\* (second\_num\_count - 1) + between\_digits \* 10 + first\_digit  
   
   print('Изменённое второе число:', second\_n)  
  
   print('\nСумма чисел:', first\_n + second\_n)

**Что оценивается**

* программа разбита на несколько функций;
* выполнены условия по организации основного тела программы.

Задача 6. Маятник

**Что нужно сделать**

Известно, что амплитуда качающегося маятника с каждым разом затухает на 8,4% от амплитуды прошлого колебания. Если качнуть маятник, то, строго говоря, он не остановится никогда, просто амплитуда будет постоянно уменьшаться до тех пор, пока мы не сочтём такой маятник остановившимся. Напишите программу, определяющую, сколько раз качнётся маятник, прежде чем он, по нашему мнению, остановится.

Программа получает на вход начальную амплитуду колебания в сантиметрах и конечную амплитуду его колебаний, которая считается остановкой маятника. Обеспечьте контроль ввода.

**Пример:**

Введите начальную амплитуду: 1

Введите амплитуду остановки: 0.1

Маятник считается остановившимся через 27 колебаний

**Что оценивается**

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 7. Яйца

**Что нужно сделать**

В рамках программы колонизации Марса компания «Спейс Инжиниринг» вывела особую породу черепах, которые, по задумке, должны размножаться, откладывая яйца в марсианском грунте. Откладывать яйца слишком близко к поверхности опасно из-за радиации, а слишком глубоко — из-за давления грунта и недостатка кислорода. Вообще, факторов очень много, но специалисты проделали большую работу и предположили, что уровень опасности для черепашьих яиц рассчитывается по формуле D = x^3 − 3x^2 − 12x + 10, где x — глубина кладки в метрах, а D — уровень опасности в условных единицах. Для тестирования гипотезы нужно взять пробу грунта на безопасной, согласно формуле, глубине.

Напишите программу, находящую такое значение глубины х, при котором уровень опасности как можно более близок к нулю. На вход программе подаётся максимально допустимое отклонение уровня опасности от нуля, а программа должна рассчитать приблизительное значение х, удовлетворяющее этому отклонению. Известно, что глубина точно больше нуля и меньше четырёх метров. Обеспечьте контроль ввода.

**Пример:**

Введите максимально допустимый уровень опасности: 0.01

Приблизительная глубина безопасной кладки: 0.732421875 м

**Что оценивается**

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 8. Сумма ряда

**Что нужно сделать**

Пользователь вводит действительное число х и точность precision. Напишите программу, которая по числу х вычисляет сумму ряда в точности до precision.

https://go.skillbox.ru/media/files/share/1637681678710.png

Операцией возведения в степень и функцией factorial пользоваться нельзя.

**Пример:**

Введите точность: 0.001

Введите x: 5

Сумма ряда =  0.2836250150891709

**Что оценивается**

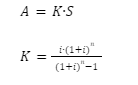
* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Задача 9. Аннуитетный платёж

**Что нужно сделать**

Кредит в сумме S млн руб., выданный на n лет под i% годовых, подлежит погашению равными ежегодными выплатами в конце каждого года, включающими процентные платежи и сумму в погашение основного долга. Проценты начисляются в один раз в год. После выплаты третьего платежа достигнута договорённость между кредитором и заёмщиком о продлении срока погашения займа на n\_2 лет и увеличении процентной ставки с момента конверсии до i\_2%. Напишите программу, которая выводит план погашения оставшейся части долга.

Используйте следующие формулы (А — размер аннуитетного платежа, его дробную часть округлите до двух знаков, то есть до копеек):



i - процент в дробном виде (6% —> 0.06)

**Пример:**  
Введите сумму кредита: 40e6  
На сколько лет выдан? 5  
Сколько процентов годовых? 6

Период: 1  
Остаток долга на начало периода: 40000000.0  
Выплачено процентов: 2400000.0  
Выплачено тела кредита: 7095856.02

Период: 2  
Остаток долга на начало периода: 32904143.98  
Выплачено процентов: 1974248.6387999998  
Выплачено тела кредита: 7521607.3812

Период: 3  
Остаток долга на начало периода: 25382536.5988  
Выплачено процентов: 1522952.195928  
Выплачено тела кредита: 7972903.824072

Остаток долга: 17409632.774728

 ====================

На сколько лет продлевается договор? 2  
Увеличение ставки до: 10

Период: 1  
Остаток долга на начало периода: 17409632.774728  
Выплачено процентов: 1740963.2774728  
Выплачено тела кредита: 3751267.5625271997

Период: 2  
Остаток долга на начало периода: 13658365.2122008  
Выплачено процентов: 1365836.52122008  
Выплачено тела кредита: 4126394.3187799198

Период: 3  
Остаток долга на начало периода: 9531970.89342088  
Выплачено процентов: 953197.0893420881  
Выплачено тела кредита: 4539033.750657911

Период: 4  
Остаток долга на начало периода: 4992937.142762969  
Выплачено процентов: 499293.71427629696  
Выплачено тела кредита: 4992937.125723703

Остаток долга: 0.017039266414940357

**Что оценивается**

* результат вывода соответствует условию;
* input содержит корректное приглашение для ввода;
* формат вывода соответствует примеру;
* вывод содержит описание результата (выведенные числа сопровождаются текстовым описанием);

Что оценивается (общее)

* Использование именованных индексов, не просто i (видео 7.2).
* Необходимо вводить правильные числа, без дополнительных действий со стороны пользователя, без использования +1 (видео 7.4).
* Домашнее задание принимается только через Fork или корректно оформленный repl.it.
* Правильно оформленный input, без пустого приветствия для ввода (видео 2.3).
* Переменные имеют значащие имена, не только a, b, c, d (видео 2.3).
* Пробелы после запятых, пробелы при бинарных операциях.
* Отсутствие пробелов после имён функций и перед скобками: “print ()”,“input ()” — неверно, “print()” — верно.
* Правильно оформлены блоки if-elif-else, отступы одинаковы во всех блоках одного уровня.

**Правильно:**

if a > 1:

  b = 3

else:

  b = 5

**Неправильно:**

If a > 1:

  b = 3

else:

    b = 5

* Переменные имеют корректные названия, и в качестве имён не используются имена встроенных функций ([список встроенных функций](https://docs.python.org/3.7/library/functions.html) — официальная документация).

Советы и рекомендации

* Выводите значение счётчика при отладке программы (видео 8.4).
* Будьте аккуратны при изменении внутри цикла чисел, которые его задают (видео 8.5).
* Помните о приоритете арифметических операций [PEP 8](https://docs.python.org/3.7/reference/expressions.html#operator-precedence)(мы вводим and, or).
* Почитайте руководство по написанию кода на Python:
  + [PEP 8](https://www.python.org/dev/peps/pep-0008/)(на английском языке),
  + [PEP 8](https://letpy.com/python-guide/pep8/) (на русском языке).
* Обратите внимание на [список встроенных функций](https://docs.python.org/3.7/library/functions.html) (официальная документация).
* Прочитайте описание конструкции [for-else.](https://docs.python.org/3/tutorial/controlflow.html#break-and-continue-statements-and-else-clauses-on-loops)
* Прочитайте описание [enumerate](https://docs.python.org/3/library/functions.html#enumerate)для исключения конструкций вида for i in range(len(array)).