Лабораторная работа №1. Ввод/вывод данных.

Философия Python (для ознакомления)

1. Красивое лучше, чем уродливое;
2. Явное лучше, чем неявное;
3. Простое лучше, чем сложное;
4. Сложное лучше, чем запутанное;
5. Плоское лучше, чем вложенное;
6. Разреженное лучше, чем плотное;
7. Читаемость имеет значение;
8. Особые случаи не настолько особые, чтобы нарушать правила;
9. При этом практичность важнее безупречности;
10. Ошибки никогда не должны замалчиваться;
11. Если не замалчиваются явно;
12. Встретив двусмысленность, отбрось искушение угадать;
13. Должен существовать один — и, желательно, только один — очевидный способ сделать что-либо;
14. Хотя он [способ] поначалу может быть и не очевиден, если вы не голландец;
15. Сейчас лучше, чем никогда;
16. Хотя никогда зачастую лучше, чем прямо сейчас;
17. Если реализацию сложно объяснить — идея плоха;
18. Если реализацию легко объяснить — идея, возможно, хороша;
19. Пространства имён — отличная вещь! Давайте будем делать их больше!

Laboratory work No. 1. Data input/output.

Philosophy of Python (for review)

1. Beautiful is better than ugly;
2. Explicit is better than implicit;
3. Simple is better than complex;
4. Complicated is better than confusing;
5. Flat is better than attached;
6. Sparse is better than dense;
7. Readability matters;
8. Special cases are not special enough to break the rules;
9. At the same time, practicality is more important than impeccability;
10. Errors should never be disguised;
11. If not explicitly hidden;
12. When faced with ambiguity, resist the temptation to guess;
13. There should be one—and preferably only one—obvious way to do anything;
14. Although the way may not be obvious at first if you are not Dutch;
15. Now is better than never;
16. Although never is often better than right now;
17. If the implementation is difficult to explain, the idea is bad;
18. If the implementation is easy to explain, the idea is probably good;
19. Namespaces are a great thing! Let's make more of them!

Писать код будем:
В среде Anaconda -> Jupyter Notebook.
Для работы также удобно использовать Google Colab: https://colab.google/ . Для этого войдите в свой аккаунт Google или зарегистрируйтесь.
Для тестирования простого кода можно использовать онлайн-редакторы:
Online Python:https://www.online-python.com/ZYsXyNdltb
или https://pythononline.kz/ -> Открыть компилятор
или https://pythononline.net/

Теория 1. **Вывод данных**

Вывод данных - команда `print()`. В скобках могут быть указаны, например, число или текст (последний в двойных или одинарных кавычках) – аргументы или параметры команды. Даже если функция не получает никаких аргументов, все равно необходимо вставлять после названия пустые скобки, что подает интерпретатору команду на исполнение функции (будет выведена пустая строка).

Комментарий в Python начинается с символа хеша # и продолжается до конца физической строки. Для добавления многострочного комментария используются тройные кавычки до и после комментария.

We will write the code:
In the program Anaconda -> Jupyter Notebook.
It is also convenient to use Google Colab for work: https://colab.google/ . To do this, log in to your Google Account or sign up.
You can use online editors to test simple code:

Online Python:https://www.online-python.com/ZYsXyNdltb
or https://pythononline.kz/ -> Open Compiler
or https://pythononline.net/

Theory 1. **Data output**
Watch the video
Data output- command `print()`. Parentheses can contain text in double or single quotes (inverted commas). Even if the function does not receive any arguments, it is still necessary to insert empty brackets after the name, which means for the interpreter to execute the function (an empty line will be output), and not just refer to it.

A comment in Python begins with the hashtag # and continues until the end of the physical line. Use triple quotes before and after a comment to add a multiline comment.

Note: 1. If the text displayed on the screen uses quotation marks, for example, double, then print () should use others, for example, single.

Примечание1. Если в тексте, выводимом на экран, используются кавычки, например, двойные, то в print() должны применяться другие, например, одинарные.

Примечание2. После вывода на экран всех данных, указанных в скобках, команда print(), по умолчанию, переводит курсор на новую строку.

Примечание3. Команда print() не может быть записана заглавными или начинаться с такой буквы, т.к. Python различает заглавные и строчные буквы.

Примечание4. Соглашение Pep 8 описывает стиль написания кода. С ним можно ознакомиться по адресу https://peps.python.org/pep-0008/

Например, обратите внимание на использование пробелов в строке, которая будет выводиться на экран с помощью функции print():

- между каждой парой слов должен быть только один пробел;
- в начале и в конце выводимой на экран строки пробелов быть не должно;
- перед запятой пробел не ставится, после – ставится один пробел.

```
# Пример 1.1.1-1) Команда print().
print("Привет!")
```

Note: 2. After displaying all the data specified in brackets, the print () command, by default, moves the cursor to a new line.

Note: 3. The print () command cannot be capitalized or begin with such a letter, because Python distinguishes between capital and lowercase letters.

Note: 4. The Pep 8 convention describes the writing style of the code. It can be found at https://peps.python.org/pep-0008/

For example, pay attention to the use of spaces in the line that will be printed to the screen using the print() function:

- there should be only one space between each pair of words;
- here should be no spaces at the beginning and at the end of the displayed line (string);
- There is no space before the comma, and one space after the comma.

```
# Example 1.1.1-1) print() command.
print("Hello!")
```

```
# Пример 1.1.1-2) Вывод текста на экран и
использование комментариев.
print('Теперь мы крутые "программисты"')
# Мы умеем писать комментарии.
'''
В том числе,
многострочные комментарии.
'''
```

Задача 1.1. Вывести на экран свое имя.
Примечание. Здесь и далее в первой строке кода указывайте номер и условие задачи как комментарий.

Задача 1.2. Вывести на экран текст:
`Здравствуйте, мой никнейм: "[ваш_никнейм]".`
Ник должен быть выведен в кавычках.
Т.е. должно получиться примерно следующее:
`Здравствуйте, мой никнейм: "КлёВыЙ Nickname"`

```
# Example 1.1.1-2) Displaying text and using
comments.
print('Now we are cool "programmers"')
# We can write comments
'''
Including
multi-line comments.
'''
```

Task 1.1. Display your name on the screen.
*Note.*Here and below, in the first line of code, indicate the number and condition of the problem as a comment.

Task 1.2. Display text on the screen:
`Hello, my nickname is "[your_nickname]"`
The nickname must be enclosed in quotes.
I.e. there should be something like this:
`Hello, my nickname is "cOOl NickName"`

Теория 2. Необязательные параметры вывода данных По умолчанию команда `print()` принимает несколько аргументов (параметров), перечисленных через запятую, и выводит их через один пробел, после чего ставит перевод строки. Но у этой команды есть и необязательные параметры. Рассмотрим параметры `sep` (separator, разделитель) и `end` (окончание). Параметр `sep` позволяет установить строку, с помощью которой будут разделены аргументы при печати. Например, чтобы аргументы выводились через запятую, нужно указать: `sep=','`. Если функция должна выводить аргументы в виде отдельных строк, можно передать параметру символ перевода каретки: `sep='\n'`	**Theory 2. Optional data output parameters** By default, the print() command accepts several arguments (parameters) separated by commas and outputs them separated by a single space, followed by a line feed. But this command also has optional parameters. Let's look at the `sep` (separator) and `end` (end) parameters. The `sep` parameter allows you to set a line that will separate the arguments when printing. For example, to output the arguments separated by commas, you need to specify: `sep=','`. If the function should output the arguments as separate lines, you can pass a carriage return sign to the parameter: sep='\n'

Если перевод строки делать не нужно и/или требуется указать специальное окончание строки, то следует явно задать значение для параметра `end`. Например, чтобы в конце строки добавить символ точки следует указать `end='.'`. В этом случае курсор не перейдет на новую строку.

```
# Пример 1.2.1-1)
print("Вывод пирамиды из чисел")
print("или с использованием параметра sep:
")
print("1","2 2","3 3 3","4 4 4 4","5 5 5 5
5",sep="\n")


# Пример 1.2.1-2)
print("Слова, разделенные одной пустой
строкой")
print("Сегодня", "прекрасный", "день",
sep="\n\n",end="!")

# Пример 1.2.1-3)
print("Слова, разделенные двумя символами:
“!” и “ ”")
print("Сегодня", "прекрасный", "день",
sep="! ")

# Пример 1.2.1-4)
print("Слова, разделенные двумя символами:
“!” и “ ”, а также символ “!” в конце
строки")
print("Сегодня", "прекрасный", "день",sep="!
",end="!")
```

If you do not need to do a line feed and/or you need to specify a special line ending, then you should explicitly set the value for the `end` parameter. For example, to add a dot sign at the end of a line, specify `end='.'`. In this case, the cursor will not move to a new line.

```
# Example 1.2.1-1)
print("The output of the pyramid of numbers")
print("1","2 2","3 3 3","4 4 4 4","5 5 5 5
5", sep="\n")


# Example 1.2.1-2)
print("words separated by one blank line")
print("Today", "wonderful", "day",
sep="\n\n",end="!")


# Example 1.2.1-3)
print("words separated by two symbols: “!”
and “ ”")
print("Today", "wonderful", "day", sep="! ")


# Example 1.2.1-4)
print("words separated by two symbols: “!”
and “ ”, and also the “!” symbol at the end
of the line")
print("Today", "wonderful", "day", sep="!
",end="!")
```

Задача 1.3. Вывести на экран день, месяц и год Вашего рождения (в каждом пункте обязательно использовать ровно одну команду `print()` с тремя соответствующими аргументами, а также параметры sep и/или end)

1) на отдельной строке каждый, например:
01
01
2001

2) через две пустых строки, например:
01

01

2001

3) на одной строке разделенные точкой, например:
01.01.2001

4) разделенные знаком «/» и с указанием «.» в конце, например:
01/01/2001.

Задача 1.4. Вывести свои фамилию, имя и отчество в одну строку, разделив символами, которые печатаются вместо

Task 1.3. Display the day, month, and year of your birth (in each item be sure to use exactly one `print()` command with three corresponding arguments, as well as the `sep and/or end` parameters)

1) each on a separate line, for example:
01
01
2001

2) separated by two blank lines, for example:
01

01

2001

3) on one line separated by a period, for example:
01.01.2001

4) separated by the "/" sign and with "." at the end, for example:
01/01/2001.

Task 1.4. Output your last name, first name, and patronymic in one line, separated by symbols that are printed instead of the digits of your variant number when the Shift key is pressed in the Russian-language keyboard layout. For example, if Ivanov Ivan Ivanovich has variant 100, then the program should output the line:

цифр номера Вашего варианта при нажатой клавише Shift в русскоязычной раскладке. Например, если у Иванова Ивана Ивановича вариант 100, то программа должна вывести строку:
`Иванов!))Иван!))Иванович`
В конце добавить восклицательный знак.
(Обязательно использовать ровно одну команду `print()` с тремя соответствующими аргументами, а также параметры sep и/или end).

Теория 3.Ввод данных с клавиатуры. Переменные. Типы данных. Арифметические операции

Ввод данных – команда `input()`.
Внутри скобок можно указать в кавычках текст, который будет выводиться перед полем ввода для пояснения.

Введенные данные могут быть присвоены некоторой ***переменной***, т.е. именованной области памяти.

Примечания.

1. Имя переменной может содержать только строчные, a-z и заглавные, A-Z, латинские буквы, цифры и символ нижнего подчеркивания (_).
2. Имя переменной не может начинаться с цифры.
3. Имя переменной принято делать осмысленным.
4. Нельзя использовать зарезервированные слова в имени переменной.
5. Язык Python чувствителен к регистру, т.е. A≠*a*.

`Ivanov!))Ivan!))Ivanovich`
Add an exclamation mark at the end.
(Be sure to use exactly one `print()` command with three corresponding arguments, as well as the `sep` and/or `end` parameters).

Theory 3. Keyboard input. Variables. Data types. Arithmetic operations

Data input – the `input()` command.
Inside the brackets, you can specify text in quotation marks that will be displayed before the input field for clarification.

The entered data can be assigned to a variable, i.e. a named memory area.

Notes.

1. The variable name can only contain lowercase, a-z and capital, A-Z, Latin letters, numbers, and the underscore (_) sign.
2. The variable name cannot begin with a number.
3. It is customary to make the variable name meaningful.
4. You cannot use reserved words in the variable name.
5. The Python language is case-sensitive, i.e. A≠a.
6. To assign a value to a variable, use the operator "=". In this case, the variable name is always indicated on the left, and the value on the right.

6. Для присваивания переменной какого-либо значения служит оператор «=». При этом имя переменной всегда указывается слева, а значение – справа.

7. При выводе переменной с помощью команды `print()` она обрамляется запятыми, когда необходимо вывести в тексте ее значение.

```
# Пример 1.3.1) Программа выясняет ФИО больного,
#год его рождения и выводит на экран сообщение
#доктору Иванову И.И. о записи на прием.
print('Представьтесь, пожалуйста.')
surname = input('Фамилия: ')
name = input('Имя: ')
patronymic = input('Отчество: ')
birth_year = input('Введите год рождения: ')
print('Здравствуйте, Иван Иванович. На сегодня к Вам записан', surname, name, patronymic, 'год рождения:', birth_year)
```

Примечание. Обратите внимание на расстановку знаков препинания.

Язык Python дает возможность работать с различными типами данных, например, `int` (целое число), `float` (число с плавающей точкой, т.е. действительное число), `bool` (логический тип), `str` (строковый тип), `list` (список), set (множество), `dict` (словарь) и др.

С числами в Python можно выполнять, например,

7. When outputting a variable using the `print()` command, it is surrounded by commas when it is necessary to output its value in the text.

```
# Example 1.3.1) The program finds out the patient's full name,
#his year of birth and displays a message on the screen
#to Dr. Ivanov I.I. about making an appointment.
print('Please introduce yourself.')
surname = input('Surname: ')
name = input('First name: ')
patronymic = input('Patronymic: ')
birth_year = input('Enter the year of birth: ')
print('Hello, Ivan Ivanovich. I have an appointment for you today', surname, name, patronymic, 'year of birth:', birth_year)
```

Note: Pay attention to the punctuation.

The Python language allows you to work with various data types, such as `int` (integer), `float` (floating point number, i.e. real number), `bool` (logical type), `str` (string type), `list` (list), `set` (set, multitude), `dict` (dictionary), etc.

With numbers in Python, you can perform, for example, the following arithmetic operations:

следующие арифметические операции:

Операции	Оператор в Python	Пример
Сложение	+	2 + 5 = 7
Вычитание	-	9 - 2 = 7
Умножение	*	4 * 3 = 12
Деление	/	12 / 3 = 4.0
Получение целой части от деления	//	7 // 4 = 1
Получение остатка от деления	%	7 % 4 = 3
Возведение в степень	**	4 ** 2 = 16

```
# Пример 1.3.2) Основные арифметические
операции.
print('2 + 2 =', 2 + 2)
print('3 – 2 =', 3 - 2)
print('5 * 5 =', 5 * 5)
print('15 / 3 =', 15 / 3)
print('25 // 3 =', 25 // 3)
print('25 % 3 = ', 25 % 3)
print('2**3 = ', 2**3)
print('Можно вычислить значение выражения:')
print('6 * 34 + 11 - (27 * 2 - 3 / 7) =', 6
* 34 +11 - (27 * 2 - 3 / 7))
```

*Примечание.*В действительных числах целая часть от дробной отделяется точкой.

Operator	Operations in Python	Example
Addition	+	2 + 5 = 7
Subtraction	-	9 - 2 = 7
Multiplication	*	4 * 3 = 12
Division	/	12 / 3 = 4.0
Getting the integer part from division	//	7 // 4 = 1
Obtaining the remainder from division	%	7 % 4 = 3
Exponentiation	**	4 ** 2 = 16

```
# Example 1.3.2) Basic arithmetic operations.
print('2 + 2 =', 2 + 2)
print('3 – 2 =', 3 - 2)
print('5 * 5 =', 5 * 5)
print('15 / 3 =', 15 / 3)
print('25 // 3 =', 25 // 3)
print('25 % 3 = ', 25 % 3)
print('2**3 = ', 2**3)
print('You can calculate the value of the
expression:')
print('6 * 34 + 11 - (27 * 2 - 3 / 7) =', 6 *
34 +11 - (27 * 2 - 3 / 7))
```

Note: In real numbers, the integer part is separated from the fractional part by a period.

Вывод действительного числа с фиксированным количеством знаков после запятой, например, с двумя:

```
print("%.2f" % x)
```

здесь % – оператор форматирования строки, .2 – требуемая точность (в данном случае 2 знака после запятой), *f* – вещественный тип данных float, x – выводимая переменная.

```
# Пример 1.3.3) Форматированный вывод десятичных чисел
divide = 365 / 7
print("%.2f" % divide)
```

Для того, чтобы *преобразовать строку в целое число*, используется команда `int()`.

Для того, чтобы *преобразовать строку в действительное число*, используется команда `float()`.

Для того, чтобы *преобразовать целое число в строку*, используется команда `str()`.

```
'''
Пример 1.3.4-1) Работа с числовыми данными.
1.  Для введенного с клавиатуры целого x вычислить значение функции f(x)=5x+8x^2 и вывести с тремя знаками после запятой.
```

Output of a real number with a fixed number of decimal places, for example, two:

```
print("%.2f" % x)
```

here % is the string formatting operator, .2 is the required precision (in this case 2 decimal places), *f* is the float data type, *x* is the output variable.

```
# Example 1.3.3) Formatted output of decimal numbers
divide = 365 / 7
print("%.2f" % divide)
```

To convert a string to an integer, use the `int()` command.

To convert a string to a real number, use the `float()` command.

To convert an integer to a string, use the `str()` command.

```
'''
Example 1.3.4-1) Working with numeric data.
1.  For an integer x entered from the keyboard, calculate the value of the function f(x)=5x+8x^2. Output the result in a format with 3 decimal places
'''
```

```
'''
x = int(input('Введи x: '))
print('f(x)=', '%.3f' %(5 * x + 8 * x**2))
```

Или иначе:

```
'''
Пример 1.3.4-2) Работа с числовыми данными.
2.  Для x = 3.1 вычислить значение функции
y(x)=5x+8x^2 и вывести с тремя знаками после
запятой.
'''
x = 3.1
print('y(x)=', '%.3f' %(5 * x + 8 * x**2))
```

```
'''
Пример 1.3.4-3) Работа с числовыми данными.
Вычислить индекс массы тела по известному
росту  в метрах и массой в килограммах.
Результат вывести в формате с 12 знаками
после запятой (ИМТ = m/h^2).
'''
h, m = float(input('h=')),
float(input('m='))
BMI = m/(h*h)
print("ИМТ=", "%.12f" % BMI) #
форматированный вывод результата с 12
знаками после запятой.
```

Примечание. Команда `float(input())` переводит

```
x = int(input('input x: '))
print('f(x)=', '%.3f' %(5 * x + 8 * x**2))
```

Or else:

```
'''
Example 1.3.4-2) Working with numeric data.
2.  For  x=3.1,  calculate  the  value  of  the
function y(x)=5x+8x^2. Output the result in a
format with 3 decimal places
'''
x = 3.1
y = 5 * x + 8 * x**2
print('y(x)=', '%.3f' %y)
```

```
'''
Example 1.3.4-3) Working with numeric data.
Calculate  the  body  mass  index  based  on  the
known  height  in  meters  and  weight  in
kilograms. Output the result in a format with
12 decimal places (BMI = m/h^2).
'''
h, m = float(input('h=')), float(input('m='))
BMI = m/(h*h)
print("BMI=", "%.12f" % BMI) #
formatted  output  of  the  result  with  12
decimal places.
```

Note. The `float(input())` command converts the string entered from the keyboard, which actually contains a number in text format, into a floating-point number format, and the `int(input())` command converts it into an integer.

введенную с клавиатуры строку, на самом деле содержащую число в текстовом формате, в формат числа с плавающей точкой, а команда `int(input())` – в целое число.

Задача 1.5. Напишите программу, запрашивающую в отдельные переменные фамилию, имя и отчество пациента, а также его основной диагноз и выводит на экран в одну строку текст, аналогичный приведенному ниже (с соблюдением всех знаков препинания и отступов), где в квадратных скобках указано, где нужно подставить введенные значения:
ФИО: [Фамилия] [Имя] [Отчество]; диагноз: [диагноз]
Например,
ФИО: Иванов Иван Иванович; диагноз: грипп
(обратите внимание на расстановку знаков препинания и пробелов)

Задача 1.6. Известно, что x=8,5 и y=7. Напишите программу, вычисляющую значение z по формуле ниже и выводящую результат с числом знаков после запятой, равным Вашему варианту.
B-1. $z=3x+7,6y^2+14$;
B-2. $z=11x^3+2y-1,2$;
B-3. $z=13x-y^9+8,4$;
B-4. $z=-8x^5+11,8y^2+4$;
B-5. $z=x+17,1y^4-11$;
B-6. $z=7x^{11}+9,9y+2$;
B-7. $z=x^7+7,6y-7$;
B-8. $z=6,6x^3+5y^2+4$;
B-9. $z=9x^2-1,1y^2+33$;

Task 1.5. Write a program that requests the patient's last name, first name, and patronymic name, as well as his primary diagnosis, in separate variables and displays on the screen in one line a text similar to the one below (with all punctuation marks and indents), where the square brackets indicate where the entered values should be substituted:
Full name: [Last name] [First name] [Patronymic name]; Diagnosis: [Diagnosis]
For example,
Full name: Ivanov Ivan Ivanovich; diagnosis: influenza
(note punctuation and spaces)

Task 1.6. It is known that x=8.5 and y=7. Write a program that calculates the value of z using the formula below and displays the result with the number of decimal places equal to your variant.

V-1. $z=3x+7,6y^2+14$;
V-2. $z=11x^3+2y-1,2$;
V-3. $z=13x-y^9+8,4$;
V-4. $z=-8x^5+11,8y^2+4$;
V-5. $z=x+17,1y^4-11$;
V-6. $z=7x^{11}+9,9y+2$;
V-7. $z=x^7+7,6y-7$;
V-8. $z=6,6x^3+5y^2+4$;
V-9. $z=9x^2-1,1y^2+33$;
V-10. $z=3,5x^4+3y^2+11$;
V-11. $z=3x+7,2y^2+14$;
V-12. $z=-1,7x^3+7y^2-15,6$;
V-13. $z=3,1x-4y^2+1,4$;

B-10. $z=3,5x^4+3y^2+11$; *B-11.* $z=3x+7,2y^2+14$; *B-12.* $z=-1,7x^3+7y^2-15,6$; *B-13.* $z=3,1x-4y^2+1,4$; *B-14.* $z=x^4+12y^2-4,8$; *B-15.* $z=2,7x+13y^5+6,9$; *B-16.* $z=-2,3x^7+2,7y^3+2,2$; *B-17.* $z=4,4x^4-4y+44$; *B-18.* $z=6,3x^2-y^3+21$; *B-19.* $z=8,9x+1,7y^2-19$; *B-20.* $z=3,3x+7,3y^9+6$.	*V-14.* $z=x^4+12y^2-4,8$; *V-15.* $z=2,7x+13y^5+6,9$; *V-16.* $z=-2,3x^7+2,7y^3+2,2$; *V-17.* $z=4,4x^4-4y+44$; *V-18.* $z=6,3x^2-y^3+21$; *V-19.* $z=8,9x+1,7y^2-19$; *V-20.* $z=3,3x+7,3y^9+6$.
Задача 1.7. Напишите программу, которая (по вводимым с клавиатуры заданным значениям)	**Task 1.7.** Write a program that (given the values entered from the keyboard)
B-1. определяет цену деления шприца, мл, если его вместимость 10 мл, а количество делений равно 20, и выводит результат с 6 знаками после запятой (цена_деления_шприца=вместимость_шприца / количество_делений).	V-1. determine the division value of a syringe, in ml, if its capacity is 10 ml and the number of divisions is 20, and outputs the result with 6 decimal places (syringe_division_value = syringe_capacity / number_of_divisions).
B-2. переводит вес в 187 фунтов в килограммы и выводит результат в формате с 3 знаками после запятой (1 фунт = 0,453592 кг);	V-2. convert the weight of 187 pounds to kilograms and outputs the result in a format with 3 decimal places (1 pound = 0.453592 kg);
B-3. выводит отношение возраста (19 лет) к длине волос (250 мм) в формате с 4 знаками после запятой.	V-3. output the ratio of age (19 years) to hair length (250 mm) in a format with 4 decimal places (age/length).
B-4. вычисляет ориентировочную норму прибавки массы тела ребенка 9 месяцев и выводит результат на экран с точностью 3 знака после запятой (масса ребенка около 1	V-4. Calculate the approximate weight gain rate for a 9-month-old child (age) and displays the result on the screen with an accuracy of 3 decimal places (the weight of a child of about 1 year of age is equal to the weight of a child of 6 months, which is about 8300 g (weight_in_6) , minus 800 g for each missing month or plus 400 g (waight_gain) for each subsequent month, i.e. for 9 months: weight_in_6 + (age - 6) · weight_gain);

года жизни равна массе тела ребенка 6 месяцев, это около 8300 г, минус 800 г на каждый недостающий месяц или плюс 400 г на каждый последующий, т.е. для 9 месяцев: 8300+400x3);	V-5. Calculate the approximate height gain rate for a 3-year-old child (age) and displays the result with an accuracy of 15 decimal places (the height of a child after one year is equal to the height of a child at 8 years, which is 130 cm (height_in_8), minus 7 cm (height_gain) for each missing year, or plus 5 cm for each subsequent year, i.e. for 3 years: height_in_8 -|8-age|· height_gain);
В-5. вычисляет ориентировочную норму прибавки роста ребенка 3 лет и выводит результат с точностью 15 знаков после запятой (рост ребенка после года равен росту ребенка в 8 лет, это 130 см, минус 7 см за каждый недостающий год, либо плюс 5 см за каждый последующий год, т.е. для 3 лет: 130-7x5);	V-6. Determine the normal blood loss during childbirth (ml) for a woman weighing 66.9 kg (m), and displays the result on the screen in 7 decimal places (normally, physiological loss is 0.5% (n) of body weight: (mxn%):100);
В-6. определяет норму кровопотери в родах (мл) для женщины массой 66,9 кг, и выводит результат на экран в формате 7 знаков после запятой (в норме физиологическая потеря составляет 0,5% от массы тела: (mxn%):100);	V-7. determine the blood loss during childbirth (ml), if the circulating blood volume (CBV) is 4999 ml, and displays the result on the screen in 11 decimal places (blood loss was 10% of the CBV: CBV:10);
В-7. определяет кровопотерю в родах (мл), если объем циркулирующей крови (ОЦК) равен 4999 мл, и выводит результат на экран в формате 11 знаков после запятой (кровопотеря составила 10% от ОЦК: ОЦК:10);	V-8. determine the shock index for a pulse of 130 and systolic pressure of 90, and displays the result on the screen in 4 decimal places (the shock index is the ratio of the pulse to the systolic pressure);
В-8. определяет шоковый индекс для пульса, равного 130, и систолического давления, равного 90, и выводит результат на экран в формате 4 знаков после запятой (шоковый индекс равен отношению пульса к систолическому давлению);	V-9. calculate the percentage of weight loss of a newborn if the baby was born weighing 3800 g, and two days later its weight was 3500 g, and displays the result on the screen in 7 decimal places (the percentage of weight loss of a newborn is the difference between the initial and current weight, divided by 100%);

В-9. вычисляет процент потери веса новорожденным, если ребенок родился с весом 3800 г, а через двое суток его вес составил 3500 г, и выводит результат на экран в формате 7 знаков после запятой (процент потери веса новорожденным – это разность между первоначальным и текущим весом, деленная на 100%);	V-10. determine the percentage of the child's body weight deficit (Dw) if its birth weight was 3700 g, and at three months its weight was 5000 g (actual weight, Aw), and displays the result on the screen in 5 decimal places (the reference point is the child's weight gain in the first month - 600 g, in the next 2 - 800 g; to determine the percentage of body weight deficit, the estimated body weight, Ew, is calculated, then the difference between the estimated and actual weights is divided by the actual weight and the result is multiplied by 100%, so
В-10. определяет процент дефицита массы (Дм) тела ребенка, если его вес при рождении составлял 3700 г., а в три месяца его масса составила 5000 г. (фактическая масса, Фм), и выводит результат на экран в формате 5 знаков после запятой (за ориентир принят прирост веса ребенка в первый месяц – 600 г, в следующие 2 – по 800 г.; для определения процента дефицита массы тела, рассчитывается ориентировочная масса тела, Ом, затем разность между ориентировочной и фактической массами делится на фактическую и результат умножается на 100%, т.о. Ом = 3700+600+2х800; Дм = |Ом - Фм|/Фм х 100);	Ew = 3700+600+2x800; Dw = |Ew - Aw|/Aw x 100); V-11. determines the minimum blood pressure a 5-year-old child should have and displays the result on the screen in 9 decimal places (the maximum blood pressure in children over one year old can be approximately determined using V.I. Molchanov's formula:
В-11. определяет, какое минимальное артериальное давление должно быть у ребенка 5 лет, и выводит результат на экран в формате 9 знаков после запятой (ориентировочно максимальное артериальное давление у детей после года можно определить с помощью формулы В.И.Молчанова: $X = 80 + 2n$, где 80 – среднее давление ребенка 1 года (в мм. рт. ст.), n - возраст ребенка; минимальное давление составляет $\frac{1}{2} - \frac{2}{3}$ Х);	X = 80 + 2n, where 80 is the average blood pressure of a 1-year-old child (in mm Hg), n is the child's age; the minimum pressure is $\frac{1}{2} - \frac{2}{3}$ X); V-12. calculate the daily caloric content of a 12-year-old child's diet and displays the result on the screen in 8 decimal places (the daily caloric content is calculated using the formula: 1000 + 100n, where n is the number of years, 1000 is the daily caloric content of a one-year-old child's diet); V-13. determine how much solvent should be taken so that 0.15 g of the substance is in 0.3 ml of the solution, if there is 0.5 g of the drug in the vial of ampicillin, and displays the result on the screen in the

В-12. рассчитывает суточную калорийность пищевого рациона ребенка 12 лет и выводит результат на экран в формате 8 знаков после запятой (суточная калорийность рассчитывается по формуле: 1000 + 100*n*, где *n* – число лет, 1000 – суточная калорийность пищевого рациона для годовалого ребенка);	format of 10 decimal places (amount_of_solvent_when_diluting_the_antibiotic = mass_of_the_antibiotic_contained_in_the_vial * amount_of_solvent_in_the_finished_medicinal_product / content_of_this_antibiotic_in_the_finished_medicinal_solution);
В-13. определяет сколько нужно взять растворителя, чтобы в 0,3 мл раствора было 0,15 г вещества, если во флаконе ампициллина находится 0,5 г лекарственного средства, и выводит результат на экран в формате 10 знаков после запятой (количество_растворителя_при_разведении_антибиотика = масса_антибиотика,_содержащегося_во_флаконе * количество_растворителя_в_готовом_лекарственном _средстве / содержание_данного_антибиотика_в_готовом_лекарственном_растворе);	V-14. calculate how many drops of the drug should be taken by the patient if 0.3 ml of an aqueous solution is prescribed, and displays the result on the screen in the format of 12 decimal places (1 ml of an aqueous solution is equal to 20 drops); V-15. calculate a single dose of a drug if 0.1 g of the drug is prescribed in 3 doses per day, and displays the result on the screen in a format with 13 decimal places (single_dose_of_a_drug = daily_dose / number_of_dose);
В-14. вычисляет, сколько капель лекарственного средства следует принимать пациенту, если назначено 0,3 мл водного раствора, и выводит результат на экран в формате 12 знаков после запятой (1 мл водного раствора равен 20 каплям);	V-16. by the given a power of 100 W of a certain device in a physiotherapy room and a voltage of 210 V, find the current in amperes and displays the result in a format with 2 decimal places (I = P / U, where I is the current in amperes; P is its power in watts; U is the voltage in volts);
В-15. рассчитывает разовую дозу препарата, если назначено 0,1 г лекарственного вещества за 3 приема в сутки, и выводит результат на экран в формате 13 знаков после запятой (разовая_доза_лекарственного_вещества = суточная_доза / количество_ приемов);	V-17. calculate a single dose of a drug if 100 mg/kg is prescribed in 4 doses for a patient weighing 75 kg, and displays the result on the screen in a format with 9 decimal places (single_dose_of_a_drug = prescribed_dose * patient_weight / number_of_dose);

B-16. по мощности в 100 Вт некоторого прибора кабинета физиотерапии и напряжению в 210 В находит силу тока в амперах и выводит результат в формате с 2 знаками после запятой (I = P / U, где I – это сила тока в амперах; P – его мощность в ваттах; U – напряжение в вольтах);	V-18. calculate the cost (value) of a trip by ambulance given the distance (distance), gasoline consumption (petrol, liters per 100 km), price per liter (price) and outputs the result in a format with 3 decimal places (value = distance · petrol · price:100);
B-17. рассчитывает разовую дозу препарата, если назначено 100мг/кг за 4 приема больному весом 75 кг, и выводит результат на экран в формате 9 знаков после запятой (разовая_доза_лекарственного_вещества = назначенная_доза * вес_больного / количество_приемов);	V-19. calculate the course dose of a drug if 0.25 g of the drug is prescribed, 3 times a day, for a month, and displays the result in a format with 5 decimal places (course_dose = daily_dose * duration_of_treatment_in_days);
B-18. рассчитывает стоимость (value) поездки на автомобиле скорой помощи при известном расстоянии (distance), расходе бензина (petrol, литров на 100 км пробега), цене за литр (price) и выводит результат в формате с 3 знаками после запятой (value = distance·petrol·price:100);	V-20. determine how many hours have passed since the recorded moment of change in the patient's condition, provided that the device recorded 2678 seconds (seconds) and displays the result in a format with 15 decimal places (hours= seconds/3600).
B-19. рассчитывает курсовую дозу лекарственного средства, если назначено 0,25 г лекарственного средства, 3 раза в день, в течение месяца, и выводит результат в формате с 5 знаками после запятой (курсовая_доза = суточная_доза * длительность_курса_лечения_в_днях);	
B-20. определяет, сколько прошло часов (hours) с зафиксированного момента изменения состояния пациента, при условии, что прибором зафиксировано 2678 прошедших секунд (seconds), и выводит результат в	

формате с 15 знаками после запятой (hours= seconds/3600).

Теория 4. Получение цифр числа

Для получения цифр числа n применяются операторы нахождения остатка от деления и деление нацело.

Последовательно находим.
Последняя цифра числа: (n % 10^1) // 10^0;
Предпоследняя цифра: (n % 10^2) // 10^1;
Предыдущая цифра: (n % 10^3) // 10^2;
.....
Вторая цифра числа: (n % 10^{n-1}) // 10^{n-2};

Theory 4. Processing digits.

Watch the video

To obtain the digits of the number n, the operators of finding the remainder of division and exact division are used .

Algorithm for obtaining digits of an n-digit number.
Last digit:(num % 10^1) // 10^0;
Penultimate digit:(num % 10^2) // 10^1;
Penultimate digit:(num % 10^3) // 10^2;
.....
Second digit:(num % 10^{n-1}) // 10^{n-2};
First digit:(num % 10^n) // 10^{n-1}.

So, if $n=n_m n_{m-1}...n_k...n_2 n_1$, then
n_k = (n % 10 ** k) // 10 ** (k-1)

```
#Example 1.4.1)Consider a program for
obtaining digits of a two-digit number 17:
num = 17
a = num % 10
b = num // 10
print(a)
print(b)
```

The output of the program will be two numbers:
7

Первая цифра числа: $(n \% 10^n) // 10^{n-1}$.

Т.о., если $n=n_m n_{m-1}...n_k...n_2 n_1$, тогда
n_k = (n % 10 ** k) // 10 ** (k-1)

```
#Пример 1.4.1) Рассмотрим программу получения цифр двузначного числа 17:
num = 17
a = num % 10
b = num // 10
print(a)
print(b)
```

Результатом выполнения программы будут два числа:
7
1

Т.о. последняя цифра числа определяется всегда как **остаток от деления** числа на 10 (% 10). Чтобы отщепить последнюю цифру от числа, необходимо разделить его **нацело** на 10 (// 10).

```
#Пример 1.4.2) Рассмотрим программу получения цифр трёхзначного числа 754:
num = 754
a = num % 10
b = (num % 100) // 10
c = num // 100
print(a)
print(b)
```

1

The last digit of a number is always defined as the remainder of dividing the number by 10 (%10). To split off the last digit from a number, you need to divide it entirely by 10 (// 10).

```
#Example 1.4.2)Consider the program for obtaining the digits of a three-digit number 754:
num = 754
a = num % 10
b = (num % 100) // 10
c = num // 100
print(a)
print(b)
print(c)
```

The output of the program will be three numbers:
4
5
7

```
#Example 1.4.3)
# 1. Write a program that determines the number of tens and ones in a two-digit number.
# The number of units is the last digit of the number, the number of tens is the first digit.
# To get the last digit of any number,
```

```
print(c)
```

Результатом выполнения программы будут три числа:

```
4
5
7
```

```
# Пример 1.4.3)
# 1. Напишите программу, определяющую число
десятков и единиц в двузначном числе.
#Число единиц – это последняя цифра числа,
число десятков – первая цифра.
# Чтобы получить последнюю цифру любого
числа,
# нужно найти остаток от деления числа на
10.
# Чтобы найти первую цифру двузначного
числа,
# нужно поделить число нацело на 10:


num = int(input())
last_digit = num % 10
first_digit = num // 10
print('Число десятков =', first_digit)
print('Число единиц =', last_digit)

# 2. Напишите программу, в которой
рассчитывается сумма цифр двузначного числа.
num = int(input())
last_digit = num % 10
first_digit = num // 10
# need to find the remainder of dividing the
number by 10.
# To find the first digit of a two-digit
number,
# need to divide the number by 10:

num = int(input())
last_digit = num % 10
first_digit = num // 10
print('Number of tens =', first_digit)
print('Number of units =', last_digit)

# 2. Write a program that calculates the sum
of the digits of a two-digit number.
num = int(input())
last_digit = num % 10
first_digit = num // 10
print('Sum of digits =', last_digit +
first_digit)

# Example 1.4.4) The program determines the
number of hundreds of mg in a drug dosage of
525 mg.
dose = 525
hundred = doze // 100
```

Task 1.8. Write a program that asks the user for the year of birth in the YY format (i.e. the last two digits) and gives a "forecast" of the possible number of his diseases during his life according to the formula:
Number_of_diseases = 5 * Number_of_tens + 10 * Number_of_units + 11

```
print('Сумма цифр =', last_digit +
first_digit)

# Пример 1.4.4) Программа определяет число
сотен мг в дозировке лекарства, равной 525
мг.
dose = 525
hundred = doze // 100
```

Задача 1.8. Напишите программу, которая запрашивает у пользователя год рождения в формате ГГ (т.е. последние две цифры) и выдает «прогноз» возможного числа его заболеваний в течение жизни по формуле:
Число_заболеваний = 5 * Число_десятков + 10 * Число_единиц + 11

Задача 1.9. Напишите программу, которая демонстрирует рост ваших знаний в наивысших оценках – выводит на экран указанный треугольник (двумя способами):
5
55
555

Перевод: Лысикова Ю.А., Антонова Ю.Н.
Translation: Lysikova Yu.A., Antonova Yu.N.

5555
55555
555555
5555555

Задача 1.10. Напишите программу, которая запрашивает у пользователя названия трех лекарственных средств, а затем выводит их в обратном порядке, каждое на отдельной строке (обязательно использовать параметры функции вывода).

Задача 1.11. Напишите программу, которая запрашивает у пользователя названия трех лекарственных средств и спецсимвол, а затем выводит фразу вида:
[препарат1][спецсимвол][препарат2][спецсимвол][препарат3] – лучшее предложение месяца!

Задача 1.12. Напишите программу, которая запрашивает у пользователя название лекарства и выводит на экран фразу:
«”[Название препарата]”! Обязательно поможет!» (без внешних кавычек).

Задача 1.13. Напишите программу, которая выводит на экран три последовательно идущих дозировки данного препарата (с вводимым с клавиатуры шагом), каждое число – на отдельной строке (для этого обязательно используйте параметры функции вызова). Первое

значение также вводится с клавиатуры, а остальные вычисляются в программе.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ.

1. «Поколение Python»: курс для начинающих // https://stepik.org/course/58852/promo (дата обращения 22.11.22).
2. Академия искусственного интеллекта для школьников https://ai- academy.ru/ (дата обращения 22.03.22).
3. Беккер, М.С. Применение математических методов в медицине: методическое пособие / М.С. Беккер. – Ессентуки, 2006. – 35 с.
4. 12. Все о Python. Программирование на Python 3 // https://all- python.ru/ (дата обращения 22.03.22)
5. Биофармация: учебник для фармацевтических вузов и факультетов / В.В. Гладышева [и др.]; под ред. В.В. Гладышева. – 2-е изд. –. Днипро: ЧМП «Экономика». – 2018. – 250 с.
6. Павленко, В. Интерактивный учебник языка Python / В. Павленко, В. Соломатин, Д.П. Кириенко, команда Pythontutor // https://pythontutor.ru/lessons/sets/(дата обращения 22.11.22).
7. Культин, Н.Б. C/C++ в задачах и примерах / Н.Б. Культин. – 2-е изд., перераб. и доп. – Спб.: БХВ-Петербург, 2011. – 368 с.
8. Медицинский справочник онлайн // http://www.makhaon.com/index.php?lng=ru&p=dict (дата обращения 22.11.22).

9. Питонтьютор // https://pythontutor.ru/ (дата обращения 22.11.22).
10. Питончик // https://pythonchik.ru/ (дата обращения 22.11.22).
11. Показатели СМАД в практической и исследовательской кардиологии // https://www.schiller.ru/profile/articles/smad/146/ (дата обращения 22.11.22).
12. Беккер, М.С. Применение математических расчетов в медицине: методическая разработка / М.С. Беккер // https://nsportal.ru/shkola/algebra/library/2012/06/09/primenenie- matematicheskikh-raschetov-v-meditsine (дата обращения 22.11.22).
13. Набиуллина С.Н. Решение расчетных задач на разведение антибиотиков: методическая разработка фрагмента занятия по математике / С.Н. Набиуллина. – Курган, 2018 // https://www.informio.ru/files/main/documents/2018/12/fragment_zanj atija_Nabiullina_SN.docx (дата обращения 22.11.22).
14. Лукашевский, С. Руководство по языку Python / С. Лукашевский // https://pyprog.pro/python/py/py_guide.html (дата обращения 22.11.22).
15. Справочник заболеваний // https://illness.docdoc.ru/ (дата обращения 22.11.22).