Лабораторная работа № 3. Циклы

Теория 3.1. Расширенные операторы присваивания

Если переменную необходимо увеличить (или уменьшить) на (или в) какое-то число раз, Python позволяет просто заменить предыдущее значение переменной на новое. Например,

```
counter = counter + 1
summa = summa % 2
```

Чтобы не приходилось писать имя переменной дважды, используют расширенные операторы присваивания. Для примеров выше:

```
counter += 1
summa %= 2
```

Примечание. При записи расширенного оператора между знаком арифметической операции и знаком равенства пробела нет. Иначе будет выведено сообщение об ошибке *SyntaxError: invalid syntax.*

Оператор	Пример использования	Эквивалент

Laboratory work № 3. Cycles

Theory 3.1. Advanced assignment operators

Watch the video

If a variable needs to be increased (or decreased) by (or in) a certain number of times, Python allows you to simply replace the previous value of the variable with the new one. For example,

```
counter = counter + 1
summa = summa % 2
```

To avoid having to write the variable name twice, advanced assignment operators are used. For the examples above:

```
counter += 1
summa %= 2
```

Note: When writing an advanced operator, there is no space between the arithmetic operation sign and the equal sign. Otherwise, the error message *SyntaxError: invalid syntax* will be displayed.

Operator	Example of use	Equivalent
+=	x += 5	x = x + 5
-=	x -= 2	x = x - 2

+=	x += 5	x = x + 5
-=	x -= 2	x = x - 2
*=	x *= 10	x = x * 10
/=	x /= 4	x = x / 4
//=	x //= 4	x = x // 4
%=	x %= 4	x = x % 4
**=	x**=3	x=x**3

```
'''
Пример 3.1.1) В ходе эксперимента два параметра изменяются следующим образом: вначале первый увеличивается на 5, затем к результату прибавляется второй параметр, после чего полученное значение умножается на второй и, наконец, вычисленный результат нацело делится на второй параметр.
'''
a, b = int(input('a: ')), int(input('b: '))
print('Увеличиваем a на 5: a += 5,')
a += 5

print('a =', a)
print('Увеличиваем новое значение a на b: a += b')
a += b
```

*=	x *= 10	x = x * 10
/=	x /= 4	x = x / 4
//=	x //= 4	x = x // 4
%=	x %= 4	x = x % 4
**=	x**=3	x=x**3

```
'''
Example 3.1.1) During the experiment, two parameters are changed as follows: first, the first is increased by 5, then the second parameter is added to the result, after which the obtained value is multiplied by the second, and finally, the calculated result is divided evenly by the second parameter.
'''
a, b = int(input('a: ')), int(input('b: '))
print('Increase a by 5: a += 5,')
a += 5
print('a =', a)
print('Increase the new value of a by b: a += b')
a += b
print('now a =', a)
print('Increase the new value of a by b times: a *= b,')
a *= b
```

```
print('теперь a =', a)
print('Увеличиваем новое значение a в b раз:
a *= b,')
a *= b
print('получаем a =', a)
print('И находим целую часть от деления a на
b: a //= b,')
a //= b
print('окончательно, a =', a)
```

Задача 3.1. Средняя температура кожи рассчитывается по формуле:
СТТ = 0,3 х Тгрудь + 0,3 х Тплечи + 0,2 х Тбедро + 0,2 х Тголень.
Соответствующие измерения производятся последовательно. Напишите программу, которая по-очереди запрашивает данные измерений, умножает на соответствующие коэффициенты и накапливает в переменной СТТ. (Используйте расширенный оператор присваивания).

Задача 3.2. Напишите программу, которая год Вашего рождения делит нацело на день рождения, а к результату прибавляет удвоенный месяц рождения. Используйте расширенный оператор присваивания. Вначале год рождения присвойте переменной g, а затем производите вычисления с этой переменной, в нее же помещайте результат. Выполняется в несколько действий. Числа с клавиатуры не вводятся.

```
print('we get a =', a)
print('And find the integer part of dividing
a by b: a //= b,')
a //= b
print('finally, a =', a)
```

Task 3.1. Average skin temperature is calculated using the formula:

CTT = 0.3 x Tchest + 0.3 x Tshoulders + 0.2 x Thips + 0.2 x Tcalf.
The corresponding measurements are taken sequentially. Write a program that requests the measurement data in turn, multiplies by the corresponding coefficients, and accumulates in the CTT variable. (Use the extended assignment operator).

Task 3.2. Write a program that divides your year of birth by your birthday and adds twice the month of birth to the result. Use the extended assignment operator. First, assign the year of birth to the variable g, then perform calculations with this variable, and place the result in it. It is performed in several steps. Numbers are not entered from the keyboard.

Теория 3.2. Цикл for

Цикл в Python задает повторяющиеся действия. Возможны две ситуации: когда требуется повторить набор операторов заранее заданное число раз (применяется цикл `for`), и случаи, когда число повторов неизвестно, но задано некоторое условие, при котором воспроизведение действий прекращается (цикл `while`).

Рассмотрим вначале цикл *for*.

Формат записи:

```
for i in range(количество_повторений):
    [тело_цикла]
```

Примечание 1. Здесь `i` – переменная цикла. Может иметь, вообще говоря, любое доступное для переменной имя.

Примечание 2. Однократное выполнение тела цикла называется *итерацией цикла.*

Примечание 3. Как и для записи условного оператора, при описании цикла *for* обязательно ставить двоеточие в конце первой строки, а все операторы, включенные в тело цикла, начинать с отступа в четыре пробела.

Theory 3.2. The for cycle

Watch the video

A cycle in Python specifies repeating actions. There are two possible situations: when it is necessary to repeat a set of statements a predetermined number of times (the `for` cycle is used), and cases when the number of repetitions is unknown, but a certain condition is specified under which the reproduction of actions stops (the `while` cycle).

Let's first consider the `for` cycle.

The format of the entry:

```
for i in range(number_of_repetitions):
    [cycle_body]
```

Note 1. Here `i` is the cycle variable. It can have, generally speaking, any name accessible to the variable.

Note 2. A single execution of the cycle body is called an iteration of the cycle.

Note 3. As with the entry of a conditional operator, when describing a for cycle, it is necessary to put a colon at the end of the first line, and all statements included in the cycle body must begin with an indent of four spaces.

```
# Example 3.2.1) Let's look at the code that
will print
# 5 times the saying of F. Bacon "Knowledge
is power":
for i in range(5):
    print('Knowledge is power')
```

```
# Пример 3.2.1) Рассмотрим код, который
распечатает
# 5 раз высказывание Ф. Бэкона "Знание -
сила":
for i in range(5):
    print('Знание - сила')
```

Цикл *for* позволяет упростить ввод данных с клавиатуры
и их однотипную обработку.

```
'''
Пример 3.2.2) Программа считывает данные 3-х
испытаний
и возводит каждое в квадрат.
'''
for i in range(3):
    num = float(input())
    print(num, '^ 2 = ', num ** 2) #
повторяются в теле цикла
print('Выполнено') # команда написана без
отступа, поэтому значение выводится 1 раз,
после окончания цикла.


'''
Пример 3.2.3) Программа считывает с
клавиатуры результаты 10 испытаний и находит
среди них наименьшее.
'''
min_res = float(input())
for i in range(10):
```

The `for` cycle allows you to simplify data entry from the keyboard and their uniform processing.

```
'''
Example 3.2.2) The program reads the data of
3 tests
and squares each.
'''
for i in range(3):
    num = float(input())
    print(num, '^ 2 = ', num ** 2) # repeated
in the body of the cycle
print('Done') # the command is written
without indentation, so the value is printed
1 time, after the end of the cycle.


'''
Example 3.2.3) The program reads the results
of 10 tests from the keyboard and finds the
smallest among them.
'''
min_res = float(input())
for i in range(10):
    res = float(input())
    if res < min_res:
        min_res = res
print('minimum result: ', min_res)
```

Task 3.3. Write a program that helps a student prepare for a lesson and repeats the specified medical term 100 times to better consolidate knowledge:

```
    res = float(input())
    if res < min_res:
        min_res = res
print('минимальный результат: ', min_res)
```

Задача 3.3. Напишите программу, которая помогает студенту подготовиться к занятию и для лучшего закрепления знаний повторяет 100 раз указанный медицинский термин:

B-1. Адинамия — уменьшение или полное прекращение деятельной активности в результате нарушения нервно-мышечного аппарата.

B-2. Акинез — отсутствие активных движений.

B-3. Брадикинезия — общая замедленность движений.

B-4. Васкулит — воспаление стенок кровеносных сосудов.

B-5. Гипостезия — понижение поверхностной чувствительности.

B-6. Дистрофия — патологическое состояние, характеризующее различные проявления расстройства питания.

B-7. Железодефицитная анемия — форма малокровия, обусловленная дефицитом железа в организме.

B-8. Заболеваемость — показатель распространения

V-1. Adynamia is a decrease or complete cessation of activity as a result of a disorder of the neuromuscular apparatus.

V-2. Akinesia is the absence of active movements.

V-3. Bradykinesia is a general slowness of movement.

V-4. Vasculitis is an inflammation of the walls of blood vessels.

V-5. Hypoesthesia is a decrease in superficial sensitivity.

V-6. Dystrophy is a pathological condition characterizing various manifestations of an eating disorder.

V-7. Iron deficiency anemia is a form of anemia caused by iron deficiency in the body.

V-8. Morbidity is an indicator of the prevalence of diseases detected and registered during the year among the population (calculated per 100, 1000, 10,000 or 100,000 inhabitants).

V-9. Injection is a method of parenteral administration of drugs or diagnostic agents into the body.

V-10. Capillaries are the thinnest vessels of the microcirculatory bed through which blood and lymph move.

болезней, выявленных и зарегистрированных в течение года среди населения (исчисляется на 100, 1000, 10 000 или 100 000 жителей). *B-9.* Инъекция — способ парентерального введения в организм лекарственных или диагностических средств. *B-10.* Капилляры — самые тонкостенные сосуды микроциркуляторного русла, по которым движется кровь и лимфа. *B-11.* Лимфа — жидкая ткань организма, содержащаяся в лимфатических сосудах и узлах. *B-12.* Мигрень — заболевание, характеризующееся приступообразной, преимущественно односторонней, головной болью, которая сопровождается вегетативными нарушениями. *B-13.* Невриты — поражения отдельных периферических нервов различной этиологии. *B-14.* Орган — обособленная часть организма, выполняющая определенные специфические функции. *B-15.* Патология — заболевание; состояние, отличающееся от нормы. B-16. Разрез — рассечение кожи или слизистой оболочки и подлежащих мягких тканей. *B-17.* Симптом — признак болезни, не свойственный	V-11. Lymph is a liquid tissue of the body contained in the lymphatic vessels and nodes. V-12. Migraine is a disease characterized by paroxysmal, predominantly one-sided, headache, which is accompanied by autonomic disorders. V-13. Neuritis is damage to individual peripheral nerves of various etiologies. V-14. An organ is a separate part of the body that performs certain specific functions. V-15. Pathology is a disease; a condition that differs from the norm. V-16. Incision is a cut in the skin or mucous membrane and underlying soft tissues. V-17. Symptom is a sign of a disease that is not characteristic of a healthy organism, used for diagnosis and prognosis of the disease. V-18. Thrombus is a compacted mass of coagulated blood or lymph that formed during life in the bloodstream or lymphatic system. V-19. Pharmacology is the science of the effect on the body of chemical compounds used to treat, prevent, and diagnose human diseases.

здоровому организму, используемый для диагностики и прогноза заболевания. *В-18.* Тромб — уплотненная масса свернувшейся крови или лимфы, образовавшаяся прижизненно в кровеносном или лимфатическом русле. *В-19.* Фармакология — наука о действии на организм химических соединений, применяемых для лечения, профилактики и диагностики болезней человека. *В-20.* Шок — патологический процесс, развивающийся вследствие расстройств нейрогуморальной регуляции, вызванных экстремальными воздействиями. **Задача 3.4.** Если введено число, большее номера Вашего варианта, то вывести на экран номер варианта столько раз, каков Ваш год рождения, иначе, - 3 раза повторить день Вашего рождения. Выводить числа в одну строку, через символ «*». **Задача 3.5.** При проведении исследования заболевания, группы испытуемых закодировали буквами A, B, C, D и E. Написать программу, которая печатает в столбец заболевания пациента с учетом рецидивов. Для вывода повторяющихся значений использовать цикл.	V-20. Shock is a pathological process that develops as a result of neurohumoral regulation disorders caused by extreme effects. **Task 3.4.** If a number greater than your variant is entered, then display the variant on the screen as many times as your year of birth, otherwise, repeat your birthday 3 times. Output numbers in one line, separated by the symbol "*". **Task 3.5.** During the study of the disease, the groups of subjects were coded with the letters A, B, C, D and E. Write a program that prints the patient's disease in the column, taking into account relapses. To output repeating values, use a cycle.

B-1.	B-2.	B-3.	B-4.	B-5.	B-6.	B-7.	B-8.	B-9.	B-10.
A B C C C C D D E	A B C D D D D D D D D E E E	A B B C D E E E E E E E E E	A A A A A B C C C D E	A B B B C D D D D E	A B C C C C C C D E E E	A A A A B C D E E E E E E	A B B B B B C C C D E	A A A A A A A B C C C D E	A A B B B B B B B B C D E

V-1.	**V-2.**	**V-3.**	**V-4.**	**V-5.**	**V-6.**	**V-7.**	**V-8.**	**V-9.**	**V-10.**
A B C C C D D E	A B C D D D D D D D D E E E	A B B B C D E E E E E E E E	A A A A A B C C C D E	A B B B C D D D D E	A B C C C C C C D E E E	A A A A B C D E E E E E E	A B B B B B C C C D E	A A A A A A A B C C C D E	A A B B B B B B B B C D E

B-11.	B-12.	B-13.	B-14.	B-15.	B-16.	B-17.	B-18.	B-19.	B-20.
A B B B C D D D D E	A B B B C C C C C C C D E	A B C C C C D E E E	A A A A A A A A B C D D D D D E	A A B B B B B B B B C D	A A A A B C D D D E	A B C D D E E E E E	A B C C C C C C C C D D D E	A B C D D D D D D D D E E E	A B C C D E E E E E E E E E

V-11.	**V-12.**	**V-13.**	**V-14.**	**V-15.**	**V-16.**	**V-17.**	**V-18.**	**V-19.**	**V-20.**
A B B B C D D D D E	A B B B C C C C C C C D E	A B C C C C D E E E	A A A A A A A A B C D D D D D E	A A B B B B B B B B C D	A A A A B C D D D E	A B C D D E E E E E E	A B C C C C C C C C D D D E	A B C D D D D D D D D E E E	A B C C D E E E E E E E E E

Теория 3.3. Изменение переменной цикла

Для задания переменных цикла традиционно используют буквы *i*, *j*, *k* или нижнее подчеркивание «_», если обращение к переменной в теле цикла не предполагается.

Рассмотрим на примере, как изменяется переменная при выполнении цикла.

```
# Пример 3.3.1) Вывод на экран переменной цикла
for i in range(5):
    print(i)
```

Примечание. Переменная цикла «пробегает» все значения из заданного диапазона. В примере выше последовательность всех целых чисел **от 0 до *n*-1** генерируется при помощи функции *range*(*n*).

```
# Пример 3.3.2) Вывод нумерованного списка пациентов.
for i in range(10):
    x = input('Введите ФИО:')
    print('Пациент №', i, x)
```

Задача 3.6. Написать программу, которая выводит на экран кубы всех целых чисел, начиная с 0, до числа Вашего рождения включительно в формате:

0 -> 0
1 -> 1
2 -> 8
3 -> 27

Theory 3.3. Changing the cycle variable

Watch the video

The letters i, j, k or the underscore "_" are traditionally used to define cycle variables if the variable is not supposed to be accessed in the cycle body.

Let's look at an example of how a variable changes during cycle execution.

```
# Example 3.3.1) Displaying the cycle variable

for i in range(5):
    print(i)
```

Note. The cycle variable "runs" through all the values from the specified range. In the example above, the sequence of all integers from 0 to n-1 is generated using the range(n) function.

```
# Example 3.3.2) Displaying a numbered list of patients.
for i in range(10):
    x = input('Enter Full Name:')
    print('Patient #', i, x)
```

Task 3.6. Write a program that displays the cubes of all integers, starting with 0, up to and including your birthday, in the format:

0 -> 0
1 -> 1
2 -> 8
3 -> 27
…

…

Задача 3.7. Напишите программу, которая выводит таблицу умножения всех цифр на номер вашего варианта. Результат выведите
в формате:

0 x [номер варианта] = 0

1 x [номер варианта] = …

Task 3.7. Write a program that displays the multiplication table of all digits by your variant. Display the result
in the format:

0 x [variant] = 0

1 x [variant] = …

Теория 3.4. Полная структура оператора for

Если необходимо, чтобы переменная цикла изменялась в иных пределах, чем описанные выше, то применяется полная форма записи функции `range` с тремя параметрами.

Формат записи:

```
for i in range(start, stop, step):
    [тело_цикла]
```

где `start` – начальное значение (по-умолчанию равен 0);
`stop` – конечное значение (всегда не включено);
`step` – шаг (по-умолчанию равен 1).

Примечание.

-функция `range()` с одним аргументом, означает, что задан параметр *stop*.
-функция `range()` с двумя аргументами, означает, что заданы *start* и *stop*.
-функция `range()` с тремя аргументами, означает, что заданы `start, stop, step`.

Примечание 1. Если первый параметр больше второго, то функция *range*() генерирует пустую последовательность.

Примечание 2. *step* может быть отрицательным, но тогда *start* должен быть больше, чем *stop*. Пустая последовательность в этом случае генерируется, если первый параметр меньше второго.

Примечание 3. Функция *range*() может генерировать только целые числа, включая отрицательные.

Theory 3.4. Full structure of the `for` operator

Watch the video

If it is necessary for the cycle variable to change within limits other than those described above, then the full form of the range function with three parameters is used.

Writing format:

```
for i in range(start, stop, step):
    [cycle_body]
```

where `start` is the initial value (defaults to 0);
`stop` is the final value (always not included);
`step` is the step (defaults to 1).

Note.

- the `range`() function with one argument means that the `stop` parameter is specified.
- the `range`() function with two arguments means that `start` and `stop` are specified.
- the `range`() function with three arguments means that `start, stop, step` are specified.

Note 1. If the first parameter is greater than the second, the range() function generates an empty sequence.

Note 2. step can be negative, but then start must be greater than stop. An empty sequence in this case is generated if the first parameter is less than the second.

Note 3. The range() function can only generate integers, including negative ones.

Note 4. The step value cannot be zero - this will result in a ValueError.

Примечание 4. Величина шага не может равняться нулю – это приведет к ошибке *ValueError*.

Примеры использования функции range()

Вызов функции	Последовательность чисел
`range(10)`	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
`range(1, 10)`	1, 2, 3, 4, 5, 6, 7, 8, 9
`range(3, 7)`	3, 4, 5, 6
`range(7, 3)`	пустая последовательность
`range(2, 15, 3)`	2, 5, 8, 11, 14
`range(9, 2, -1)`	9, 8, 7, 6, 5, 4, 3
`range(3, 10, -2)`	пустая последовательность

```
# Пример 3.4.1) Примеры работы цикла for.
print('range(1,11)')
print('i: ',end=' ')
for i in range(1,11):
    print(i,end=' ')

print(end='\n\n')
print('range(-5,55)')
```

Examples of using the range() function

Calling the function	Sequence of numbers
`range(10)`	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
`range(1, 10)`	1, 2, 3, 4, 5, 6, 7, 8, 9
`range(3, 7)`	3, 4, 5, 6
`range(7, 3)`	empty sequence
`range(2, 15, 3)`	2, 5, 8, 11, 14
`range(9, 2, -1)`	9, 8, 7, 6, 5, 4, 3
`range(3, 10, -2)`	empty sequence

```
# Example 3.4.1) Examples of the for cycle.
  print('range(1,11)')
  print('i: ',end=' ')
  for i in range(1,11):
      print(i,end=' ')

  print(end='\n\n')
  print('range(-5,55)')
  print('i: ',end=' ')
  for i in range(-5,55):
```

```
print('i: ',end=' ')
for i in range(-5,55):
    print(i,end=' ')

print(end='\n\n')
print('range(11,1)')
print('i: ',end=' ')
for i in range(11,1):
    print(i,end=' ')
print(end='-пустая последовательность')

print(end='\n\n')
print('range(1,11,2)')
print('i: ',end=' ')
for i in range(1,11,2):
    print(i,end=' ')

print(end='\n\n')
print('range(-5,55,10)')
print('i: ',end=' ')
for i in range(-5,55,10):
     print(i,end=' ')

print(end='\n\n')
print('range(11,1,-1)')
print('i: ',end=' ')
for i in range(11,1,-1):
    print(i,end=' ')

print(end='\n\n')
print('range(55,-5,-10)')
print('i: ',end=' ')
for i in range(55,-5,-10):
```

```
    print(i,end=' ')

print(end='\n\n')
print('range(11,1)')
print('i: ',end=' ')
for i in range(11,1):
    print(i,end=' ')
print(end='-empty sequence')

print(end='\n\n')
print('range(1,11,2)')
print('i: ',end=' ')
for i in range(1,11,2):
    print(i,end=' ')

print(end='\n\n')
print('range(-5,55,10)')
print('i: ',end=' ')
for i in range(-5,55,10):
     print(i,end=' ')

print(end='\n\n')
print('range(11,1,-1)')
print('i: ',end=' ')
for i in range(11,1,-1):
    print(i,end=' ')

print(end='\n\n')
print('range(55,-5,-10)')
print('i: ',end=' ')
for i in range(55,-5,-10):
    print(i,end=' ')
```

```
    print(i,end=' ')


print(end='\n\n')
print('range(1,11,-1)')
print('i: ',end=' ')
for i in range(1,11,-1):
    print(i,end=' ')
print(end='-пустая последовательность')
```

Задача 3.8. Напишите программу, которая выводит числа из промежутка от дня Вашего рождения до года Вашего рождения включительно, с шагом, равным Вашему варианту, в одну строку, через символ «;».

Задача 3.9. Напишите программу, которая выводит в порядке убывания все целые числа из промежутка (используйте отрицательный шаг цикла):

B-1. [–10; 10]
B-2. (–5; 4]
B-3. [–8; 7)
B-4. (–9; 11)
B-5. [–3; 18]
B-6. (–6;15]
B-7. [–11;5)
B-8. (–4; 12)
B-9. [–7; 14]
B-10. (–11; 3]
B-11. [–9; 10)
B-12. (–13; 16)

```
print(end='\n\n')
print('range(1,11,-1)')
print('i: ',end=' ')
for i in range(1,11,-1):
    print(i,end=' ')
print(end='-empty sequence')
```

Task 3.8. *Write a program that outputs numbers from the interval from your birthday to the year of your birth inclusive, with a step equal to your variant, in one line, separated by the symbol ";".*

Task 3.9. *Write a program that outputs all integers from the interval in decreasing order (use a negative cycle step):*

V-1. [–10; 10]
V-2. (–5; 4]
V-3. [–8; 7)
V-4. (–9; 11)
V-5. [–3; 18]
V-6. (–6;15]
V-7. [–11;5)
V-8. (–4; 12)
V-9. [–7; 14]
V-10. (–11; 3]
V-11. [–9; 10)
V-12. (–13; 16)
V-13. [–17; 5]
V-14. (–14; 4]

B-13. [–17; 5]
B-14. (–14; 4]
B-15. [–7; 25)
B-16. (–23; 8)
B-17. [–12; 22]
B-18. (–34; 3]
B-19. [–11; 55)
B-20. (–44; 4)

V-15. [–7; 25)
V-16. (–23; 8)
V-17. [–12; 22]
V-18. (–34; 3]
V-19. [–11; 55)
V-20. (–44; 4)

Теория 3.5. Цикл while

Цикл *while* нужен тогда, когда заранее не известно число итераций.

Формат записи:

```
while [условие]:
```

Theory 3.5. While cycle

Watch the video

The `while` cycle is needed when the number of iterations is not known in advance.

Writing format:

```
while [condition]:
    [cycle_body]
```

(while the condition before the mandatory colon is met, execute the commands written with an indent of four spaces, as soon as the condition is met, exit the cycle).

Note 1. If the condition is false before the cycle, then no iterations will be performed.

Note 2. If the condition cannot be met, an infinite cycle will result. The cycle body must provide for changing the condition to false, some stop value.

Note 3. The condition can contain logical operations `and, or, not.`

```
'''
Example 3.5.1) The program will print 5 times
the statement by F. Bacon "Knowledge is
power" using the while cycle:
'''

i = 0
while i < 5:
   print('Knowledge is power')
   i += 1
```

[тело_цикла]

(пока условие, стоящее до обязательного двоеточия, выполняется, – делай команды, записанные с отступом в четыре пробела, как только условие выполнено – выход из цикла).

Примечание 1. Если условие ложно до цикла, то не будет произведено ни одной итерации.

Примечание 2. Если условие не может быть выполнено, получится бесконечный цикл. В теле цикла должно быть предусмотрено изменение условия на ложное, некоторое стоп-значение.

Примечание 3. Условие может содержать логические операции *and, or, not.*

```
'''
Пример 3.5.1) Программа распечатает 5 раз
высказывание
Ф. Бэкона "Знание - сила" при помощи цикла
while:
'''
i = 0
while i < 5:
    print('Знание - сила')
    i += 1

'''
Пример 3.5.1)' Программа распечатает 5 раз
высказывание
Ф. Бэкона "Знание - сила" при помощи цикла
while(True):
'''
```

```
'''
Example 3.5.1)' The program will print 5
times the statement
by F. Bacon "Knowledge is power" using the
while(True) cycle:
'''
i = 0
while True:
   if i < 5:
     print('Knowledge is power')
     i += 1
   else:
     break
```

Task 3.10. Write a program that duplicates on the screen the names of drugs entered from the keyboard until "Enough!" is entered.

Task 3.11. Real numbers are entered from the keyboard. Write a program that displays their squares until 0 is entered.

Task 3.12. Write a program that numbers and displays all lines entered from the keyboard until "@&@" is entered (without external quotes). Consider the cases: while [condition] and while True.

```
i = 0
while True:
    if i < 5:
        print('Знание - сила')
        i += 1
    else:
        break
```

Задача 3.10. Напишите программу, которая дублирует на экране названия вводимых с клавиатуры лекарственных препаратов, пока не будет введено «Хватит!».

Задача 3.11. С клавиатуры вводятся действительные числа. Написать программу, которая выводит на экран их квадраты, пока не введен 0.

Задача 3.12. Написать программу, которая нумерует и выводит на экран все введенные с клавиатуры строки, до тех пор, пока не введено «@&@» (без внешних кавычек). Рассмотреть случаи: while [условие] и while True.

Теория 3.6. Использование циклов для элементарных подсчетов

Для написания кода нахождения количества каких-либо значений, суммы или произведения в цикле существует несколько особенностей:

Theory 3.6. Using cycles for basic calculations

Watch the video

There are several features for writing code for finding the number of any values, sum, or product in a cycle:

- when accumulating any values in a variable during a cycle, this variable must first be assigned an initial value (before executing the cycle body).
- if a quantity or sum is being counted, the initial value of the counter variable is usually 0, unless otherwise stated in the condition.
- if a product is being counted, the initial value is 1, unless otherwise required.

1. Counting the number of iterations for a certain condition.

```
#Example 3.6.1) the program reads 10 values and determines how many of them are greater than the entered n.
n = float(input('Enter n: '))
counter = 0
for i in range(10):
   num = int(input('Enter a number: '))
   if num > n:
      counter += 1
print('There were entered', counter, 'values greater than', n)
```

Note. To count the number, we increase the counter by 1 at each iteration that satisfies the specified condition.

-при накоплении в переменной каких-либо значений в ходе цикла данной переменной нужно предварительно присвоить начальное значение (до выполнения тела цикла).

-если подсчитывается количество или сумма, то начальное значение переменной-счетчика обычно 0, если в условии не сказано иное.

-если подсчитывается произведение, то за начальное значение берется 1, если не требуется иное.

1.Подсчет количества итераций по некоторому условию.

```
#Пример 3.6.1) программа считывает 10 значений и определяет, сколько из них больше введенного n.
n = float(input('Введите n: '))
counter = 0
for i in range(10):
    num = int(input('Введите число: '))
    if num > n:
        counter += 1
print('Было введено', counter, 'значений, больших', n)
```

Примечание. Для подсчета количества увеличиваем счетчик на 1 при каждой удовлетворяющей заданному условию итерации.

```
'''
Пример 3.6.2) Программа-помощник считает количество
'''
Example 3.6.2) The assistant program counts the number of
medications prescribed by the doctor. First, it asks
how many appointments are prescribed, asks the patient to enter "+"
if the medication has been taken and "-" if the patient has missed the appointment.
Then, if “+” is entered, it displays on the screen which
dose of this medicine it is and how much is left,
if “–” it prints the text “Don’t forget to take the medicine
next time!”, otherwise – “I didn’t understand, repeat.”
As soon as the prescription is completed, it displays the corresponding entry.
'''

n = int(input("Enter the number of doses: "))
counter = 0 #set the initial value of the counter
while counter != n:
   tag = input("+ or -")
   if tag == "+":
       counter += 1 #increment the counter when the condition is met
       print("You took your medicine", counter, "times, remaining: ", n - counter, "times")
```

```
приемов лекарств по назначению врача. Вначале запрашивает
сколько приемов назначено, просит больного ввести «+»,
если лекарство выпито и «-», если больной пропустил прием.
Затем, если введен «+», то выводит на экран, какой
это по счету прием данного средства и сколько осталось,
если «-» печатает текст «Не забудьте выпить лекарство в
следующий раз!», иначе - «Не понял, повтори.». Как только
назначение выполнено, выводит соответствующую запись.
'''

n = int(input('Введите количество приемов: '))
counter = 0 #задаем начальное значение счетчика
while counter != n:
    tag = input()
    if tag == '+':
          counter += 1 # увеличиваем счетчик при выполнении условия
              print('Вы приняли лекарство', counter, 'раз, осталось: ', n - counter, 'раз')

    elif tag == '-':
       elif tag == "-":
          print("Don't forget to take your medicine next time!")
       else:
          print("I didn't get it, repeat it.")
print("Treatment completed.")
```

2. Calculating the sum in a cycle based on a certain condition.

```
'''
Example 3.6.3) To determine the magnitude of the spread of temperature fluctuations (t) of intensive care patients during the day (measured at 1-hour intervals), it is necessary to determine its average daily value. To do this,
the sum of the input values is found in the cycle and divided by the number of measurements.
'''

sum = 0
for i in range(24): # 24 temperature measurements are taken per day
   t = float(input())
   sum += t
average_t = sum / 24
print('The average daily temperature was: ', '%.1f' %average_t)
```

```
        print('Не забудьте выпить лекарство в
следующий раз!')
    else:
        print('Не понял, повторите.')
print('Лечение окончено.')
```

2.Подсчет суммы в цикле по некоторому условию.

```
'''
Пример 3.6.3) Для определения величины
разброса колебаний
температуры (t) реанимационных больных в
течение суток
(измеряется с интервалом в 1 час), требуется
определить ее среднесуточное значение. Для
этого в цикле находится
сумма вводимых показателей и делится на
количество измерений.
'''

sum = 0
for i in range(24): # за сутки производится
24 измерения температуры
    t = float(input())
    sum += t
average_t = sum / 24
print('Среднесуточная температура составила:
', '%.1f' %average_t)




'''
```

```
'''
Example 8.6.3)’ Analog of the previous
example for while:
'''
sum = 0
counter = 0
while counter < 24:
   t = float(input('Enter the temperature:
'))
   sum += t # calculate the sum of
measurements
   counter += 1 # increase the hour counter
average_t = sum / counter
print('The average temperature was: ', '%.1f'
%average_t)
```

Note. To calculate the sum, at each iteration that satisfies the given condition, we increase the counter by a certain number.

```
'''
Example 3.6.4) The program calculates the sum
of the results of temperature measurements of
intensive care patients entered from the
keyboard during the day (t), until a negative
number is entered. And then it finds their
average value (there is no analogue for for -
the number of values entered is not known in
advance).
'''
t = float(input('Enter the value: '))
sum = 0
counter = 0
```

```
Пример 8.6.3)’ Аналог предыдущего примера для while:
'''
sum = 0
counter = 0
while counter < 24:
    t = float(input('Введите температуру: '))
    sum += t # подсчитываем сумму измерений
    counter += 1 # увеличиваем счетчик часов
average_t = sum / counter
print('Средняя температура составила: ', '%.1f'
%average_t)
```

Примечание. Для подсчета суммы при каждой удовлетворяющей заданному условию итерации увеличиваем счетчик на некоторое число.

```
'''
Пример 3.6.4) Программа подсчитывает сумму вводимых с
клавиатуры результатов измерений температуры реанимационных больных в течение суток (t), пока не введено отрицательное число. И затем находит их среднее значение (аналога для for нет – количество вводимых значений заранее не известно).
'''
t = float(input('Введите значение: '))
sum = 0
counter = 0
while t >= 0:
    sum += t # подсчитываем сумму значений
    counter += 1 # count the number of measurements
    t = float(input('Enter value: '))
average_t = sum / counter
print('Average temperature was: ', '%.1f' %average_t)
```

3. Calculating the product in a cycle according to some condition.

```
'''
Example 3.6.5) The program counts the number of ways to arrange objects in a laboratory cabinet if the order of their arrangement is not important (Number of permutations P_n=n!)
'''
n = int(input('Enter a number '))
P_n = 1
for i in range(2, n+1):
    P_n *= i
print(P_n)
```

Note. To calculate the product, at each iteration that satisfies the given condition, we increase the counter by a certain number of times.

Task 3.13. Write a program (use the extended assignment operator) that reads 10 integers from the keyboard and determines:

```
        counter += 1 # считаем количество измерений
    t = float(input('Введите значение: '))
average_t = sum / counter
print('Среднеяя температура составила: ', '%.1f' %average_t)
```

3.Подсчет произведения в цикле по некоторому условию.

```
'''
Пример 3.6.5) Программа подсчитывает число способов расстановки предметов в лабораторном шкафу, если порядок их расположения не важен (Число перестановок P_n=n!)
'''
n = int(input('Введите число '))
P_n = 1
for i in range(2, n+1):
    P_n *= i
print(P_n)
```

Примечание. Для подсчета произведения при каждой удовлетворяющей заданному условию итерации увеличиваем счетчик в некоторое число раз.

Задача 3.13. Написать программу (используйте расширенный оператор присваивания), которая считывает с клавиатуры 10 целых чисел и определяет:

В-1. сколько из них больше номера вашего варианта и сколько – нечетных.

V-1. how many of them are greater than the number of your variant and how many are odd.

V-2. how many of them are even and how many are less than the number of your variant.

V-3. how many of them are equal to the number of your variant and how many are not evenly divisible by it.

V-4. how many of them are divisible by your variant and how many are greater than twice your variant.

V-5. how many of them are not divisible by your variant and how many are less than three times your variant.

V-6. how many of them are less than or equal to two times your variant and how many are divisible by it.

V-7. how many of them are not divisible by two times your variant and how many are greater than or equal to your variant.

V-8. how many of them are in the range [0; 8] and how many are divisible by your variant.

V-9. how many of them are not divisible by three times your variant and how many are in the interval (0; 9).

V-10. how many of them are in the interval [0; 10) and how many are divisible by your variant.

B-2. сколько из них четных и сколько – меньше номера вашего варианта.	V-11. how many of them are divisible by your variant and how many are in the range [0; 11].
B-3. сколько из них равны номеру вашего варианта и сколько нацело не делится на него.	V-12. how many of them are not divisible by the number of your variant multiplied by five and how many are not more than 12.
B-4. сколько из них нацело делится на номер вашего варианта и сколько – больше удвоенного номера вашего варианта.	V-13. how many of them are not less than 13 and how many are not divisible by twice the number of your variant.
B-5. сколько из них нацело не делится на номер вашего варианта и сколько – меньше утроенного номера вашего варианта.	V-14. how many of them are divisible by half of the number of your variant and how many are in the interval (0; 14].
B-6. сколько из них меньше или равны удвоенному номеру вашего варианта и сколько – нацело делится на него.	V-15. how many of them are equal to four times the number of your variant and how many are divisible by 15.
B-7. сколько из них нацело не делится на удвоенный номер вашего варианта и сколько – больше или равно номеру варианта.	V-16. how many of them are not divisible by the number of your variant and how many are in the range [0; 16].
B-8. сколько из них лежит в диапазоне [0; 8] и сколько – нацело делится на номер вашего варианта.	V-17. how many of them are divisible by twice the number of your variant and how many are in the interval [0: 17)
B-9. сколько из них нацело не делится на утроенный номер вашего варианта и сколько – лежит в интервале (0; 9).	V-18. how many of them are in the interval (0; 18] and how many are not divisible by a third of your variant.
	V-19. how many of them are in the range [0; 19] and how many are divisible by your variant.
	V-20. how many of them are not equal to three times your variant and how many are in the interval [0; 20).

В-10. сколько из них лежит в интервале [0; 10) и сколько – делится нацело на номер вашего варианта.	**Task 3.14.** Write a program that calculates
В-11. сколько из них делится нацело на номер вашего варианта и сколько – лежит в диапазоне [0; 11].	V-1. the sum of even integers entered from the keyboard until a number greater than 10*[your variant] is entered.
В-12. сколько из них не делится нацело на пятикратно увеличенный номер Вашего варианта и сколько – не больше 12.	V-2. the number of integers other than the variant entered from the keyboard until the difference between the entered number and your variant is negative.
В-13. сколько из них не меньше 13 и сколько – не делится нацело на удвоенный номер вашего варианта.	V-3. the sum of all negative integers from the set entered from the keyboard until its modulus exceeds 5*[your variant].
В-14. сколько из них делится нацело на половину от номера вашего варианта и сколько – лежит в интервале (0; 14].	V-4. the product of integers different from zero, entered from the keyboard until a number multiple of the variant is entered.
В-15. сколько из них равно учетверенному номеру вашего варианта и сколько – делится нацело на 15.	V-5. the sum of integers divisible by the variant without remainder, entered from the keyboard until a negative number is entered.
В-16. сколько из них нацело не делится на номер вашего варианта и сколько – лежит в диапазоне [0; 16].	V-6. the sum of quotients of integer division by the variant, different from 10, of integers entered until the modulus of the difference between the entered number and the variant exceeds the variant.
В-17. сколько из них делится нацело на удвоенный номер вашего варианта и сколько – лежит в интервале [0: 17)	V-7. the number of integers different from three times the variant, entered from the keyboard until an odd number is entered.
В-18. сколько из них лежит в интервале (0; 18] и сколько – не делится нацело на треть номера вашего варианта.	V-8. the product of positive integers entered from the keyboard until the modulus of this product exceeds the variant raised to the fifth power.

B-19. сколько из них лежит в диапазоне [0; 19] и сколько – делится нацело на номер вашего варианта.	V-9. the number of integers not exceeding the variant, entered from the keyboard until a number is entered that is evenly divisible by a third of the variant.
B-20. сколько из них не равно утроенному номеру вашего варианта и сколько – лежит в интервале [0; 20).	V-10. the sum of all integers entered from the keyboard that are evenly divisible by the variant until the entered value coincides with the quotient of the integer division of the variant by 3.
Задача 3.14. Написать программу, которая подсчитывает	
B-1. сумму четных целых чисел, вводимых с клавиатуры до тех пор, пока не введено число, превышающее 10*[номер вашего варианта].	V-11. the sum of all remainders from dividing positive integers by twice the variant until the difference between the entered value and the variant becomes three times the variant.
B-2. количество целых чисел, отличных от номера варианта, вводимых с клавиатуры до тех пор, пока разность введенного числа с номером вашего варианта не окажется отрицательной.	V-12. the product of integers whose square is greater than your variant, entered from the keyboard until 0 is entered.
B-3. сумму всех целых отрицательных чисел из множества вводимых с клавиатуры до тех пор, пока ее модуль не превысит 5*[номер вашего варианта].	V-13. the number of integers whose remainder from dividing by your variant does not exceed five, entered from the keyboard until the square of your variant is entered.
B-4. произведение целых чисел, отличных от нуля, вводимых с клавиатуры до тех пор, пока не введено число, кратное номеру варианта.	V-14. the sum of the squares of all integers whose cube is greater than the cube of your variant, entered from the keyboard until a negative number is entered.
B-5. сумму целых чисел, делящихся без остатка на номер варианта, вводимых с клавиатуры до тех пор, пока не введено отрицательное число.	V-15. the number of non-negative integers entered from the keyboard until the modulus of the difference between the entered number and the variant exceeds the square of the variant.
B-6. сумму частных от целочисленного деления на номер варианта, отличных от 10, целых чисел, вводимых до	

тех пор, пока модуль разности вводимого числа и номера варианта не превысит номера варианта.	V-16. the product of all negative integers entered from the keyboard until the modulus of this product exceeds 100*[your variant].
В-7. количество отличных от утроенного номера варианта целых чисел, вводимых с клавиатуры до тех пор, пока не введено нечетное число.	V-17. the product of all integers whose fifth power does not exceed the square of your variant, entered from the keyboard until 0 is entered.
В-8. произведение положительных целых чисел, вводимых с клавиатуры до тех пор, пока модуль этого произведения не превысит номера варианта, возведенного в пятую степень.	V-18. the sum of all even integers entered from the keyboard until a number is entered whose cube does not exceed three times your variant taken with the opposite sign.
В-9. количество не превышающих номера варианта целых чисел, вводимых с клавиатуры до тех пор, пока не введено число, нацело делящееся на треть номера варианта.	V-19. the number of all integers whose square is less than their cube, entered from the keyboard until the entered value exceeds 7*[your variant].
В-10. сумму всех введенных с клавиатуры целых чисел, нацело делящихся на номер варианта до тех пор, пока введенное значение не совпадет с частным от целочисленного деления номера варианта на 3.	V-20. the sum of the cubes of all integers that are multiples of your variant, entered from the keyboard until a number greater than 1,000,000 is entered. **Task 3.15.** Write a program that calculates the sum of all the divisors of your variant multiplied by 10.
В-11. сумму всех остатков от деления положительных целых чисел на удвоенный номер варианта до тех пор, пока разность между введенным значением и номером варианта не станет в три раза больше номера варианта.	**Task 3.16.** A patient needs to take 5 different medications 3 times a day for a month. The patient has decided to swallow them in a different order each time. Write a program that determines whether there are enough ways of drug intake (i.e., whether the number of permutations of 5 tablets is enough for the prescribed number of doses).
В-12. произведение целых чисел, квадрат которых больше номера вашего варианта, вводимых с клавиатуры до тех пор, пока не будет введен 0.	

B-13. количество целых чисел, остаток от деления которых на номер вашего варианта не превышает пяти, вводимых с клавиатуры до тех пор, пока не будет введен квадрат номера вашего варианта. *B-14.* сумму квадратов всех целых чисел, куб которых больше куба номера вашего варианта, вводимых с клавиатуры до тех пор, пока не введено отрицательное число. *B-15.* количество целых неотрицательных чисел, вводимых с клавиатуры до тех пор, пока модуль разности введенного числа и номера варианта не превысит квадрата номера варианта. *B-16.* произведение всех целых отрицательных чисел, вводимых с клавиатуры до тех пор, пока модуль этого произведения не превысит 100*[номер вашего варианта]. *B-17.* произведение всех целых чисел, пятая степень которых не превышает квадрата номера вашего варианта, вводимых с клавиатуры до тех пор, пока не введен 0. *B-18.* сумму всех четных целых чисел, вводимых с клавиатуры до тех пор, пока не будет введено число, куб которого не превышает утроенного номера вашего варианта, взятого с противоположным знаком. *B-19.* количество всех целых чисел, квадрат которых меньше их куба, вводимых с клавиатуры до тех пор, пока	***Theory 3.7. Break and continue operators*** The `break` operator allows you to task the cycle, if necessary. `'''`

введенное значение не превысит 7*[номер вашего варианта].

B-20. сумму кубов всех целых чисел, кратных номеру вашего варианта, вводимых с клавиатуры до тех пор, пока не будет введено число, превышающее 1.000.000.

Задача 3.15. Написать программу, которая вычисляет сумму всех делителей номера вашего варианта, умноженного на 10.

Задача 3.16. Пациенту необходимо выпивать по 5 различных лекарств 3 раза в день в течение месяца. Больной решил каждый раз проглатывать их в разном порядке. Напишите программу, которая определяет, существует ли достаточное число вариантов приема лекарств (т.е. числа перестановок из 5 таблеток достаточно для назначенного количества приемов).

```
Example 3.7.1) You need to ask the user 10
times
"Do you know that?.." and print in response
to the input the saying
of F. Bacon "Knowledge is power", but if the
word "know" is entered,
then task the execution of the cycle.
'''

for i in range(10):
   print(i+1,'- Do you know that?..')
   reply = input()
   if reply == ('know' or 'I know' or 'I
KNOW'):
         break
   print('Knowledge is power')
```

The `continue` operator allows you to skip part of the cycle, if necessary, and go to its next iteration.

```
'''
Example 3.7.2) It is required to ask
the user 10 times "Do you know that?.."
and output in response to the input the
statement by F. Bacon
"Knowledge is power", but skip
the output of the phrase at step 5:
'''

for i in range(10):
   if i == 4:
```

Теория 3.7. Операторы break и continue

Оператор `break` позволяет, при необходимости, прервать цикл.

```
'''
Пример 3.7.1) Требуется 10 раз
поинтересоваться у
пользователя «А ты знаешь, что?..» и
вывести в ответ на
ввод высказывание Ф. Бэкона "Знание –
сила", но если
введено слово «знаю», то прервать
выполнение цикла.
'''
for i in range(10):
    print(i+1,'- А ты знаешь, что?..')
    reply = input()
    if reply == ('знаю' or 'Знаю' or
'ЗНАЮ'):
        continue
    print(i+1,'- Do you know that?..')
    reply = input()
    print('knowledge is power')
```

Note. The complete structure of the `for` and `while` cycles is as follows:

```
for i in range(number_of_repetitions):
    cycle_body
else:
action_else_if_not_there_was_task
```

and

```
while condition:
    code_block
else:
action_else_if_not_there_was_task
```

```
'''
Example 3.7.3) In the conditions of example
3.6.4),
if the entered temperature is from 0 oC to 35
oC or more than 40 oC, the program outputs
"Urgent help required!" and ends the cycle,
if there was no such temperature, it
additionally outputs:

"It's good that there were no complications!"
'''
t = float(input('Enter temperature: '))
sum = 0
```

```
        break
    print('Знание - сила')
```

Оператор `continue` позволяет, при необходимости, пропустить часть цикла и перейти к следующей его итерации.

```
'''
Пример 3.7.2) Требуется 10 раз
поинтересоваться у
пользователя «А ты знаешь, что?..» и
вывести в  ответ
на ввод высказывание Ф. Бэкона "Знание -
сила", но на 5
шаге вывод фразы пропустить:
'''
for i in range(10):
    if i == 4:
        continue
    print(i+1,'- А ты знаешь, что?..')
    reply = input()
    print('знание - сила')
```

Примечание. Полная структура циклов `for` и `while` имеет вид:

```
for i in range(количество_повторений):
    тело_цикла
else:
    действия_иначе,_если_не_было_break
```

```
counter = 0
while t >= 0:
   if 0 <= t < 35 or t > 45:
      print('Urgent help required!')
      break
   sum += t # calculate the sum of
measurements
   counter += 1 # count the number of
measurements
   t = float(input('Enter temperature: '))

else:
   average_t = sum / counter
   print('Average temperature was: ', '%.1f'
%average_t)
   print()
   print('Good thing there were no
complications!')
```

Task 3.17. Write a program using a for cycle that reads 10 integers sequentially and sums them until it finds a multiple of your variant or sums them all.

Task 3.18. Write a program that finds the product of all integers, starting with 1, except your variant, until the word "stop" is entered from the keyboard.

Task 3.19. A student takes a test until he gets a "pass". The number of possible retakes is equal to your variant. Write a program that

```
и
while условие:
    блок_кода
else:
    действия_иначе,_если_не_было_break


'''
Пример 3.7.3) В условиях примера 3.6.4),
если введена
температура от 0 °C до 35 °C или больше
40°C, программа
выводит «Требуется срочная помощь!» и
завершает цикл,
если такой температуры не было, выводит
дополнительно:
«Хорошо, что не было осложнений!»
'''
t = float(input('Введите температуру: '))
sum = 0
counter = 0
while t >= 0:
    if 0 <= t < 35 or t > 45:
        print('Требуется срочная помощь!')
        break
    sum += t # подсчитываем сумму измерений

    counter += 1 # считаем количество
измерений
    t = float(input('Введите температуру:
'))
else:
    average_t = sum / counter
```

displays "Hurray! Pass!" as soon as the student passes the subject, and "You are in the army now!" otherwise.

Theory 3.8. math Module

Python has many ready-made solutions collected in libraries (modules) that can be connected to your program. One of them is

```
    print('Средняя температура составила:
', '%.1f' %average_t)
    print()
    print('Хорошо, что не было
осложнений!')
```

Задача 3.17. Написать программу, использующую цикл for, которая последовательно считывает 10 целых чисел и суммирует их до тех пор, пока не обнаружит число, кратное номеру вашего варианта или не просуммирует их все.

Задача 3.18. Написать программу, которая находит произведение всех целых чисел, начиная с 1, кроме номера вашего варианта, пока с клавиатуры не будет введено слово «stop».

Задача 3.19. Студент сдает зачет, пока не получит «зачтено». Число возможных пересдач равно номеру Вашего варианта. Написать программу, которая выводит «Ура! Зачет!», как только студент сдал предмет, и «You are in army now!» – в противном случае.

the math module, which includes various functions for working with real numbers.

Connecting the `math` module:

```
import math
```

or

```
from math import *
```

Note. In the first case, you need to access the module each time you perform calculations. The second way is convenient when you need to use the module functions frequently.

Some functions and constants of the math module:

round(*x*)	rounds the number x to the nearest even number, if the fractional part of x is 0.5, then – to the nearest even number
round(*x, n*)	rounds the number x to n places after the period
floor(*x*)	rounds number x down
ceil(*x*)	rounds number x up
sqrt(*x*)	$\sqrt{x}$
pow(*x, n*)	x^n
log(*x*)	$\ln x$
log10(*x*)	$\lg x$
log(*x, b*)	$\log_b x$

Теория 3.8. Модуль math

В языке Python существует множество готовых решений, собранных в библиотеки (модули), которые можно подключить к своей программе. Одним из них является модуль `math`, включающий различные функции для работы с действительными числами.

Подключение модуля `math`*:*

```
import math
```

или

```
from math import *
```

Примечание. В первом случае, при вычислениях необходимо каждый раз обращаться к модулю. Второй вариант удобен, когда требуется часто использовать функции модуля.

factorial(*n*)	*n*!
degrees(*x*)	converts radians to degrees
radians(*x*)	converts degrees to radians
cos(*x*)	cos x
sin(*x*)	sin x
tan(*x*)	tgx
acos(*x*)	$arccosx$
asin(*x*)	$arcsinx$
atan(*x*)	$arctgx$
pi	π=3.141592653589793
e	e=2.718281828459045

Note 1. The functions of this module described above: `int()`, `float()`, `abs()`, `min()`, `max()`, as well as the `round()` function and constants are built-in and do not require connection.

Note 2. Instead of the `pow(x,n)` function, you can use: `x**n`.

Note 3. If you need to use several specific functions, you can connect only them:

```
from math import sqrt, sin
```

Then when accessing them, you do not need to specify the name of the module.

```
'''
```

Некоторые функции и константы модуля math:

round(*x*)	округляет число *x* до ближайшего целого, если дробная часть *x* равна 0,5, то – до ближайшего четного числа.
round(*x, n*)	округляет число *x* до *n* знаков после точки
floor(*x*)	округляет число *x* вниз
ceil(*x*)	округляет число *x* вверх
sqrt(*x*)	$\sqrt{x}$
pow(*x, n*)	x^n
log(*x*)	$\ln x$
log10(*x*)	$\lg x$
log(*x, b*)	$\log_b x$
factorial(*n*)	*n*!
degrees(*x*)	переводит радианы в градусы
radians(*x*)	переводит градусы, в радианы
cos(*x*)	$\cos x$
sin(*x*)	$\sin x$
tan(*x*)	tgx
acos(*x*)	$arccosx$

```
Example 3.8.1) The program calculates the
square root of two and rounds the resulting
number up and down to the nearest integer
(connecting the math module in the first way)
'''

import math
x = math.sqrt(2)
y = math.ceil(x)
z = math.floor(x)
print('root of 2: ',x)
print('rounding up: ',y)
print('rounding down: ',z)


'''
Example 3.8.2) The program calculates the
square root of two and rounds the resulting
number up and down to the nearest integer
(connecting the math module in the second
way)
'''

from math import *
x = sqrt(2)
y = ceil(x)
z = floor(x)
print('root of 2: ',x)
print('rounding up: ',y)
print('round down: ',z)
```

asin(*x*)	$arcsinx$
atan(*x*)	$arctgx$
pi	π=3.141592653589793
e	*e*=2.718281828459045

Примечание 1. Уже описанные выше функции данного модуля: *int*(), *float*(), `abs()`, `min()`, `max()`, а также функция `round()` и константы являются встроенными и не требуют подключения.

Примечание 2. Вместо функции `pow(x,n)` можно использовать: `x**n`.

Примечание 3. Если нужно использовать несколько конкретных функций, то можно подключить только их:

```
from math import sqrt, sin
```

Тогда при обращении к ним не нужно указывать название модуля.

```
'''
Пример 3.8.1) Программа вычисляет корень
квадратный из двух и округляет полученное
число до ближайшего целого числа вверх и
вниз (подключение модуля math первым
способом)
'''
import math
x = math.sqrt(2)
y = math.ceil(x)
z = math.floor(x)
print('корень из 2: ',x)
print('округление вверх: ',y)
print('округление вниз: ',z)
```

Task 3.20. Write a program that calculates a trigonometric expression as many times as your answer number, each time asking for the angle in degrees. To convert an angle entered in degrees to an angle in radians, use the `radians()` function.

V-1. $cosx + ctg^2x$;

V-2. $ctgx - sin^2x$;

V-3. $sin^3x + tg^2 2x$;

V-4. $cosx^2 + ctgx$;

V-5. $arcsinx + cosx$;

V-6. $sinx + arctgx$;

V-7. $cos3x - ctg5x$;

V-8. $cos^3x + sin4x$;

V-9. $sin15x + ctg^3x$;

V-10. $cosx - cos^2 5x$;

V-11. $3cos3x + 2sin^2x$;

V-12. $ctg8x + cos^5x$;

V-13. $sinx + 13tg^{10}x$;

V-14. $cos^2x + ctg6x$;

V-15. $sin5x^5 - cosx$;

V-16. $cos4x - ctg^2 4x$;

V-17. $tg^2x + ctg2x$;

```
'''
Пример 3.8.2) Программа вычисляет корень
квадратный из двух и округляет полученное
число до ближайшего целого числа вверх и
вниз (подключение модуля math вторым
способом)
'''
from math import *
x = sqrt(2)
y = ceil(x)
z = floor(x)
print('корень из 2: ',x)
print('округление вверх: ',y)
print('округление вниз: ',z)
```

Задача 3.20. Напишите программу, которая вычисляет тригонометрическое выражение столько раз, каков Ваш номер варианта, каждый раз, запрашивая величину угла в градусах. Для перевода угла, введенного в градусах, в угол в радианах используйте функцию *radians*().

B-1. $cosx + ctg^2x$;

B-2. $ctgx - sin^2x$;

B-3. $sin^3x + tg^2 2x$;

B-4. $cosx^2 + ctgx$;

B-5. $arcsinx + cosx$;

V-18. $sin9x + 9tgx^9$;

V-19. $cos3x + 4sin^3x$;

V-20. $cos^2x + 2tg2x$;

Перевод: Лысикова Ю.А., Антонова Ю.Н.

Translation: Lysikova Yu.A., Antonova Yu.N.

B-6. $sinx + arctgx$;	
B-7. $cos3x - ctg5x$;	
B-8. $cos^3x + sin4x$;	
B-9. $sin15x + ctg^3x$;	
B-10 $cosx - cos^2 5x$;	
.	
B-11 $3cos3x + 2sin^2x$;	
.	
B-12 $ctg8x + cos^5x$;	
.	
B-13 $sinx + 13tg^{10}x$;	
.	
B-14 $cos^2x + ctg6x$;	
.	
B-15 $sin5x^5 - cosx$;	
.	
B-16 $cos4x - ctg^2 4x$;	
.	
B-17 $tg^2x + ctg2x$;	
.	
B-18 $sin9x + 9tgx^9$;	
.	
B-19 $cos3x + 4sin^3x$;	
.	
B-20 $cos^2x + 2tg2x$;	
.	

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).