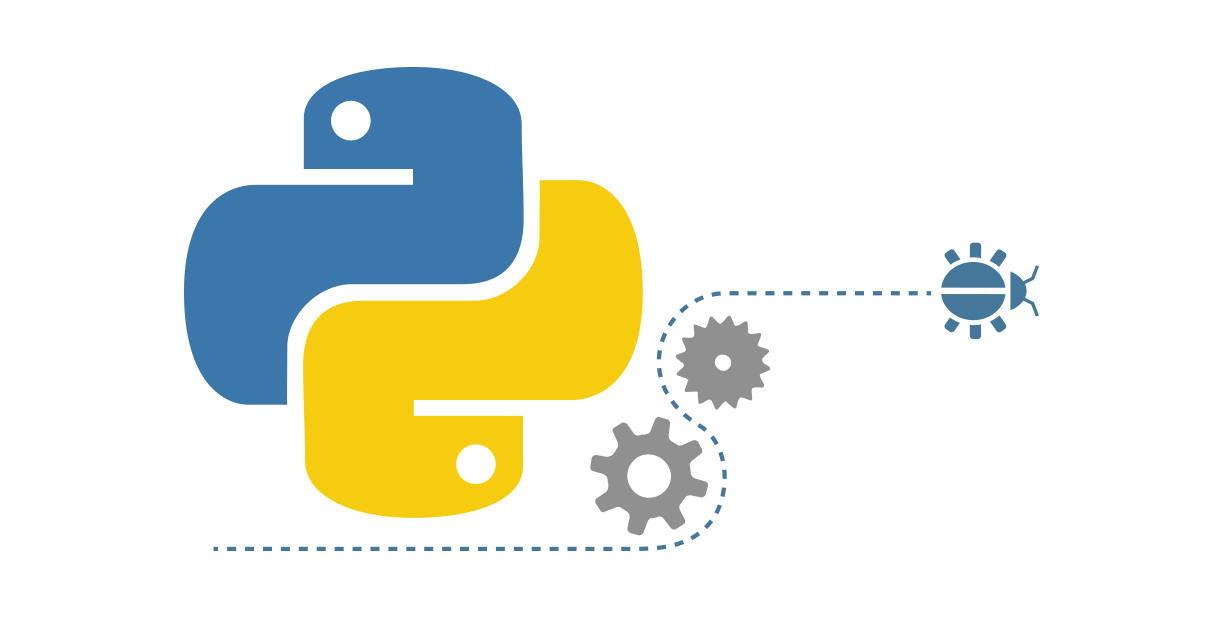
**Программалау есептер жинағы**

**PYTHON**



Введение.

Python — один из самых популярных языков программирования в мире. Python имеет предельно простой синтаксис, на нем легко писать, его легко читать, а больше и желать трудно. Другой позитивной новостью для тех, кто собрался освоить этот язык, будет нескончаемое количество справочной литературы: книг, сайтов, платных и бесплатных курсов, готовых шаблонов и исходников. На всех стадиях самообразования и работы вы не будете испытывать недостатка в современных и актуальных информационных и развлекательных ресурсах, посвящённых языку Python. Еще один позитивный момент всеобщей доступности Python — множество доступных сред разработки, сервисов и фреймворков. У вас не составит труда найти качественный и бесплатный продукт для работы дома, в офисе и в дороге. Вам не придётся тратить драгоценное время, изобретая очередной велосипед — всё уже сделано до вас и для вас. Просто оглянитесь вокруг: Youtube, Quora, Reddit, DropBox — всё это создано с использованием Python. Развитие Python позволяет сегодня создавать сложный код, не затрачивая на это много временных и «строчных» ресурсов. Ну и самое главное — программисты со знанием Python до сих пор очень востребованы на рынке труда. В России и ближнем Зарубежье прямо сейчас разыскивается около [2,5 тысяч](http://russia.trud.com/?q=python) представителей этой профессии, при этом среди них немало джуниоров. Одно у языка Python не отнять: он был крут 25 лет назад, остался крут и по сей день.   
(Статья из сайта <https://geekbrains.ru/posts/why_love_python>)

**1. Мәліметтерді енгізу және шығару**  
Питонда мәндерді шығару үшін print() функциясы қолданылады. Дөңгелек жақшалардың ішінде үтір арқылы біз шығарғымыз келетін мәндерді жазамыз. Мысалы, бірнеше есептеулерді орындайтын бағдарлама:

*print(5 + 10)*

*print(3 \* 7, (17 - 2) \* 8)*

*print(2 \*\* 16) # две звёздочки означают возведение в степень*

*print(37 / 3) # один слэш — это деление с ответом-дробью*

*print(37 // 3) # два слэша считают частное от деления нацело*

*# это как операция div в других языках*

*print(37 % 3) # процент считает остаток от деления нацело*

*# это как операция mod в других языках*

Для ввода данных в программу мы используем функцию input(). Она считывает одну строку.

*a = input()*

*b = input()*

*s = a + b*

*print(s)*

**Бағдарлама 57-ні шығарады, бірақ шынайы өмірде 5 + 7 нәтижесі 12 болуы керек еді.**  
Бұл жағдай Питон үшінші жолда екі санды емес, екі жолды (мәтінді) "қосқандықтан" орын алды. Питонда екі жолды қосу дегеніміз — бірінші жолдың соңына екінші жолды жалғау дегенді білдіреді.

Әрбір объект белгілі бір типке жатады.

* Мәтіндер ("5", "7") str типіне жатады.
* Бүтін сандар (5, 7) int типіне жатады.
* Бөлшек сандар (мысалы, 5.7) float типіне жатады.

Міне, дұрыс жұмыс істейтін бағдарлама мысалы, ол екі санды оқып, олардың қосындысын шығарады:

*a = int(input())*

*b = int(input())*

*s = a + b*

*print(s)*

Задачи

1.1. Напишите программу, которая считывает три числа и выводит их сумму. Каждое число записано в отдельной строке.

1.2. Напишите программу, которая считывает длины двух катетов в прямоугольном треугольнике и выводит его площадь. Каждое число записано в отдельной строке.

1.3. n школьников делят k яблок поровну, неделящийся остаток остается в корзинке. Сколько яблок достанется каждому школьнику? Сколько яблок останется в корзинке? Программа получает на вход числа n и k и должна вывести искомое количество яблок (два числа).

1.4. Дано число n. С начала суток прошло n минут. Определите, сколько часов и минут будут показывать электронные часы в этот момент. Программа должна вывести два числа: количество часов (от 0 до 23) и количество минут (от 0 до 59). Учтите, что число n может быть больше, чем количество минут в сутках.

1.5. Напишите программу, которая приветствует пользователя, выводя слово Hello, введенное имя и знаки препинания по образцу: Hello, Harry!

1.6. Напишите программу, которая считывает целое число и выводит текст, аналогичный приведенному в примере (пробелы важны!):  
Входные данные: 154

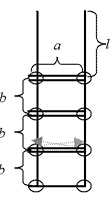
Выходные данные:   
The next number for the number 154 is 155.

The previous number for the number 154 is 153.

1.7. В школе решили набрать три новых математических класса. Так как занятия по математике у них проходят в одно и то же время, было решено выделить кабинет для каждого класса и купить в них новые парты. За каждой партой может сидеть не больше двух учеников. Известно количество учащихся в каждом из трёх классов. Сколько всего нужно закупить парт, чтобы их хватило на всех учеников? Программа получает на вход три натуральных числа: количество учащихся в каждом из трех классов.

1.8. Обувная фабрика собирается начать выпуск элитной модели ботинок. Дырочки для шнуровки будут расположены в два ряда, расстояние между рядами равно aa, а расстояние между дырочками в ряду bb. Количество дырочек в каждом ряду равно NN. Шнуровка должна происходить элитным способом “наверх, по горизонтали в другой ряд, наверх, по горизонтали и т.д.” (см. рисунок). Кроме того, чтобы шнурки можно было завязать элитным бантиком, длина свободного конца шнурка должна быть ll. Какова должна быть длина шнурка для этих ботинок?

Программа получает на вход четыре натуральных числа aa, bb, ll и NN - именно в таком порядке - и должна вывести одно число - искомую длину шнурка.



### 2. Условия

условная инструкция в Питоне имеет следующий синтаксис:

*if Условие:*

*Блок инструкций 1*

*else:*

*Блок инструкций 2*

Для выделения блока инструкций, относящихся к инструкции if или else, в языке Питон используются отступы. Все инструкции, которые относятся к одному блоку, должны иметь равную величину отступа, то есть одинаковое число пробелов в начале строки. Рекомендуется использовать отступ в 4 пробела и не рекомендуется использовать в качестве отступа символ табуляции.

Операторы сравнения

Как правило, в качестве проверяемого условия используется результат вычисления одного из следующих операторов сравнения:

**<** Меньше.

**>** Больше.

**<=** Меньше или равно.

**>=** Больше или равно.

**==** Равенство.

**!=** Неравенство.

Операторы сравнения в Питоне можно объединять в цепочки (в отличии от большинства других языков программирования, где для этого нужно использовать логические связки), например, a == b == c или 1 <= x <= 10.

Иногда нужно проверить одновременно не одно, а несколько условий. Например, проверить, является ли данное число четным, можно при помощи условия (n % 2 == 0) (остаток от деления n на 2 равен 0), а если необходимо проверить, что два данных целых числа n и m являются четными, необходимо проверить справедливость обоих условий: n % 2 == 0и m % 2 == 0, для чего их необходимо объединить при помощи оператора and (логическое И): n % 2 == 0 and m % 2 == 0.

В Питоне существуют стандартные логические операторы: логическое И, логическое ИЛИ, логическое отрицание.

### Каскадные условные инструкции. В такой конструкции условия if, ..., elif проверяются по очереди, выполняется блок, соответствующий первому из истинных условий. Если все проверяемые условия ложны, то выполняется блок else, если он присутствует.

### Задачи.

### 2.1. Даны два целых числа. Выведите значение наименьшего из них. (доп. Даны три целых числа. Выведите значение наименьшего из них.)

2.2. В математике функция sign(x) (знак числа) определена так:   
*sign(x) = 1, если x > 0,   
sign(x) = -1, если x < 0,   
sign(x) = 0, если x = 0.*

Для данного числа x выведите значение sign(x). Эту задачу желательно решить с использованием каскадных инструкций if... elif... else.

2.3. Заданы две клетки шахматной доски. Если они покрашены в один цвет, то выведите слово YES, а если в разные цвета — то NO. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки.

2.4. Дано натуральное число. Требуется определить, является ли год с данным номером високосным. Если год является високосным, то выведите YES, иначе выведите NO. Напомним, что в соответствии с григорианским календарем, год является високосным, если его номер кратен 4, но не кратен 100, а также, если он кратен 400.

2.5. Даны три целых числа. Определите, сколько среди них совпадающих. Программа должна вывести одно из чисел: 3 (если все совпадают), 2 (если два совпадает) или 0 (если все числа различны).

### 2.6. «Ход ладьи». Шахматная ладья ходит по горизонтали или вертикали. Даны две различные клетки шахматной доски, определите, может ли ладья попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести YES, если из первой клетки ходом ладьи можно попасть во вторую или NO в противном случае.

### 2.7. «Ход короля». Шахматный король ходит по горизонтали, вертикали и диагонали, но только на 1 клетку. Даны две различные клетки шахматной доски, определите, может ли король попасть с первой клетки на вторую одним ходом. Программа получает на вход четыре числа от 1 до 8 каждое, задающие номер столбца и номер строки сначала для первой клетки, потом для второй клетки. Программа должна вывести YES, если из первой клетки ходом короля можно попасть во вторую или NO в противном случае.

### 2.8. «Ход слона». Шахматный слон ходит по диагонали. Даны две различные клетки шахматной доски, определите, может ли слон попасть с первой клетки на вторую одним ходом.

### 2.9. «Ход ферзя». Шахматный ферзь ходит по диагонали, горизонтали или вертикали. Даны две различные клетки шахматной доски, определите, может ли ферзь попасть с первой клетки на вторую одним ходом.

### 2.10. «Ход коня». Шахматный конь ходит буквой “Г” — на две клетки по вертикали в любом направлении и на одну клетку по горизонтали, или наоборот. Даны две различные клетки шахматной доски, определите, может ли конь попасть с первой клетки на вторую одним ходом.

### 2.11. Шоколадка имеет вид прямоугольника, разделенного на n×m долек. Шоколадку можно один раз разломить по прямой на две части. Определите, можно ли таким образом отломить от шоколадки часть, состоящую ровно из k долек. Программа получает на вход три числа: n, m, k и должна вывести YES или NO.

### 2.12. Яша плавал в бассейне размером *N × M* метров и устал. В этот момент он обнаружил, что находится на расстоянии *x* метров от одного из длинных бортиков (не обязательно от ближайшего) и *y* метров от одного из коротких бортиков. Какое минимальное расстояние должен проплыть Яша, чтобы выбраться из бассейна на бортик? Программа получает на вход числа *N, M, x, y*. Программа должна вывести число метров, которое нужно проплыть Яше до бортика.

### 3. Вычисления

Для целых чисел определены операции +, -, \* и \*\*. Операция деления / для целых чисел возвращает вещественное число (значение типа float). Также функция возведения в степень возвращает значение типа float, если показатель степени — отрицательное число.

Но есть и специальная операция целочисленного деления, выполняющегося с отбрасыванием дробной части, которая обозначается // (она соответствует операции div в Паскале). Она возвращает целое число: целую часть частного. Другая близкая ей операция − это операция взятия остатка от деления, обозначаемая % (она соответствует операции mod в Паскале).

### Преобразование действительных чисел к целому производится с округлением в сторону нуля, то есть int(1.7) == 1, int(-1.7) == -1.

Для проведения вычислений с действительными числами язык Питон содержит много дополнительных функций, собранных в библиотеку (модуль), которая называется math.

Для использования этих функций в начале программы необходимо подключить математическую библиотеку, что делается командой

import math

Например, пусть мы хотим округлять вещественные числа до ближайшего целого числа *вверх*. Соответствующая функция ceil от одного аргумента вызывается, например, так: math.ceil(x) (то есть явно указывается, что из модуля math используется функция ceil). Вместо числа x может быть любое число, переменная или выражение.

### Задачи.

### 3.1. Дано натуральное число. Выведите его последнюю цифру.

3.2. Длина Московской кольцевой автомобильной дороги —109 километров. Байкер Вася стартует с нулевого километра МКАД и едет со скоростью v километров в час. На какой отметке он остановится через t часов?

Программа получает на вход значение v и t. Если v>0, то Вася движется в положительном направлении по МКАД, если же значение v<0, то в отрицательном.

Программа должна вывести целое число от 0 до 108 — номер отметки, на которой остановится Вася.

### 3.3. Дано положительное действительное число X. Выведите его дробную часть.

### 3.4. Дано положительное действительное число X. Выведите его первую цифру после десятичной точки.

3.5. В некоторой школе занятия начинаются в 9:00. Продолжительность урока — 45 минут, после 1-го, 3-го, 5-го и т.д. уроков перемена 5 минут, а после 2-го, 4-го, 6-го и т.д. — 15 минут.

Дан номер урока (число от 1 до 10). Определите, когда заканчивается указанный урок.

Выведите два целых числа: время окончания урока в часах и минутах.

### 3.6. За день машина проезжает n километров. Сколько дней нужно, чтобы проехать маршрут длиной m километров? Программа получает на вход числа n и m.

### 3.7. Пирожок в столовой стоит a рублей и b копеек. Определите, сколько рублей и копеек нужно заплатить за n пирожков. Программа получает на вход три числа: a, b, n, и должна вывести два числа: стоимость покупки в рублях и копейках.

3.8. Даны значения двух моментов времени, принадлежащих одним и тем же суткам: часы, минуты и секунды для каждого из моментов времени. Известно, что второй момент времени наступил не раньше первого. Определите, сколько секунд прошло между двумя моментами времени.

Программа на вход получает три целых числа: часы, минуты, секунды, задающие первый момент времени и три целых числа, задающих второй момент времени.

Выведите число секунд между этими моментами времени.

3.9. Улитка ползет по вертикальному шесту высотой h метров, поднимаясь за день на a метров, а за ночь, спускаясь на b метров. На какой день улитка доползет до вершины шеста?

Программа получает на вход натуральные числа h, a, b.

Программа должна вывести одно натуральное число. Гарантируется, что a>b.

3.10. Дано натуральное число. Найдите число десятков в его десятичной записи.

3.11. Дано трехзначное число. Найдите сумму его цифр.

3.12. Дано два числа *a* и *b*. Выведите гипотенузу треугольника с заданными катетами.

3.13. С начала суток прошло H часов, M минут, S секунд (0 ≤ H < 12, 0 ≤ M < 60, 0 ≤ S < 60). По данным числам H, M, S определите угол (в градусах), на который повернулаcь часовая стрелка с начала суток и выведите его в виде действительного числа.

3.14. С начала суток часовая стрелка повернулась на угол в α градусов. Определите сколько полных часов, минут и секунд прошло с начала суток, то есть решите задачу, обратную предыдущей. Запишите ответ в три переменные и выведите их на экран.

3.15. Процентная ставка по вкладу составляет P процентов годовых, которые прибавляются к сумме вклада. Вклад составляет X рублей Y копеек. Определите размер вклада через год.

Программа получает на вход целые числа P, X, Y и должна вывести два числа: величину вклада через год в рублях и копейках. Дробная часть копеек отбрасывается.

**4. Цикл for**

Для повторения цикла некоторое заданное число раз n можно использовать цикл for вместе с функцией range:

*for i in range(4): # равносильно инструкции for i in 0, 1, 2, 3:*

*# здесь можно выполнять циклические действия*

*print(i)*

*print(i \*\* 2)*

*# цикл закончился, поскольку закончился блок с отступом*

*print('Конец цикла')*

Функция range может также принимать не один, а два параметра. Вызов range(a, b) означает, что индексная переменная будеть принимать значения от a до b - 1, то есть первый параметр функции range, вызываемой с двумя параметрами, задает начальное значение индексной переменной, а второй параметр — первое значение, которое индексная переменная принимать **не будет**. Если же a≥b, то цикл не будет выполнен ни разу. Например, для того, чтобы просуммировать значения чисел от 1 до n можно воспользоваться следующей программой:

*sum = 0*

*n = 5*

*for i in range(1, n + 1):*

*sum += i*

*print(sum)*

### Настройка функции print()

По умолчанию функция print() принимает несколько аргументов, выводит их через пробел, после чего ставит перевод строки. Это поведение можно изменить, используя именованные параметры sep (разделитель) и end (окончание).

Задачи

4.1. Даны два целых числа A и B (при этом A ≤ B). Выведите все числа от A до B включительно.

4.2. Даны два целых числа A и В. Выведите все числа от A до B включительно, в порядке возрастания, если A < B, или в порядке убывания в противном случае.

4.3. Дано 10 целых чисел. Вычислите их сумму. Напишите программу, использующую наименьшее число переменных.

4.4. Дано несколько чисел. Вычислите их сумму. Сначала вводите количество чисел N, затем вводится ровно N целых чисел. Какое наименьшее число переменных нужно для решения этой задачи?

4.5. Факториалом числа n называется произведение 1 × 2 × ... × n. Обозначение: n!.

По данному натуральному n вычислите значение n!. Пользоваться математической библиотекой math в этой задаче запрещено.

4.6. По данному натуральному n ≤ 9 выведите лесенку из n ступенек, i-я ступенька состоит из чисел от 1 до i без пробелов.

4.7. Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Найдите ее, зная номера оставшихся карточек.

Дано число N, далее N − 1 номер оставшихся карточек (различные числа от 1 до N). Программа должна вывести номер потерянной карточки.

Массивами и аналогичными структурами данных пользоваться нельзя.

5. Строки

Узнать количество символов (длину строки) можно при помощи функции len.

Любой другой объект в Питоне можно перевести к строке, которая ему соответствует. Для этого нужно вызвать функцию str().Перевод строки в число осуществляется функцией int().Срез (slice) — извлечение из данной строки одного символа или некоторого фрагмента подстроки или подпоследовательности.

Есть три формы срезов. Самая простая форма среза: взятие одного символа строки, а именно, S[i] — это срез, состоящий из одного символа, который имеет номер i.

Срез с двумя параметрами: S[a:b] возвращает подстроку из b - a символов, начиная с символа c индексом a, то есть до символа с индексом b, не включая его. Если задать срез с тремя параметрами S[a:b:d], то третий параметр задает шаг, как в случае с функцией range, то есть будут взяты символы с индексами a, a + d, a + 2 \* d и т. д.

#### Метод count

Подсчитывает количество вхождений одной строки в другую строку. Простейшая форма вызова S.count(T)  возвращает число вхождений строки T внутри строки S. При этом подсчитываются только непересекающиеся вхождения, например:

*print('Abracadabra'.count('a'))*

*# вернёт 4*

*print(('a' \* 10).count('aa'))*

*# вернёт 5*

При указании трех параметров S.count(T, a, b), будет выполнен подсчет числа вхождений строки T в срезе S[a:b].

Задачи.

5.1. Дана строка.

Сначала выведите третий символ этой строки.

Во второй строке выведите предпоследний символ этой строки.

В третьей строке выведите первые пять символов этой строки.

В четвертой строке выведите всю строку, кроме последних двух символов.

В пятой строке выведите все символы с четными индексами (считая, что индексация начинается с 0, поэтому символы выводятся, начиная с первого).

В шестой строке выведите все символы с нечетными индексами, то есть, начиная со второго символа строки.

В седьмой строке выведите все символы в обратном порядке.

В восьмой строке выведите все символы строки через один в обратном порядке, начиная с последнего.

В девятой строке выведите длину данной строки.

5.2. Дана строка, состоящая из слов, разделенных пробелами. Определите, сколько в ней слов. Используйте для решения задачи метод count.

5.3. Дана строка, состоящая ровно из двух слов, разделенных пробелом. Переставьте эти слова местами. Результат запишите в строку и выведите получившуюся строку.

При решении этой задачи не стоит пользоваться циклами и инструкцией if.

5.4. Дана строка, в которой буква h встречается минимум два раза. Удалите из этой строки первое и последнее вхождение буквы h, а также все символы, находящиеся между ними.

5.5. Дана строка. Замените в этой строке все цифры 1 на слово one.

5.6. Дана строка. Удалите из нее все символы, чьи индексы делятся на 3.

### 6. Цикл while

При выполнении цикла while сначала проверяется условие. Если оно ложно, то выполнение цикла прекращается и управление передается на следующую инструкцию после тела цикла. После тела цикла можно написать слово else: и после него блок операций, который будет выполнен *один раз* после окончания цикла, когда проверяемое условие станет неверно. Смысл появляется только вместе с инструкцией break. Если во время выполнения Питон встречает инструкцию break внутри цикла, то он сразу же прекращает выполнение этого цикла и выходит из него. При этом ветка else исполняться не будет. Разумеется, инструкцию break осмыленно вызывать только внутри инструкции if, то есть она должна выполняться только при выполнении какого-то особенного условия. Другая инструкция управления циклом — continue (продолжение цикла). Если эта инструкция встречается где-то посередине цикла, то пропускаются все оставшиеся инструкции до конца цикла, и исполнение цикла продолжается со следующей итерации.

Задачи.

6.1. По данному целому числу N распечатайте все квадраты натуральных чисел, не превосходящие N, в порядке возрастания.

6.2. Дано целое число, не меньшее 2. Выведите его наименьший натуральный делитель, отличный от 1.

6.3. По данному натуральному числу N найдите наибольшую целую степень двойки, не превосходящую N. Выведите показатель степени и саму степень.

6.4. В первый день спортсмен пробежал *x* километров, а затем он каждый день увеличивал пробег на 10% от предыдущего значения. По данному числу *y*определите номер дня, на который пробег спортсмена составит не менее *y*километров.

Программа получает на вход действительные числа *x* и *y* и должна вывести одно натуральное число.

6.5. Последовательность состоит из натуральных чисел и завершается числом 0. Определите значение наибольшего элемента последовательности.

6.6. Последовательность состоит из натуральных чисел и завершается числом 0. Определите индекс наибольшего элемента последовательности. Если наибольших элементов несколько, выведите индекс первого из них. Нумерация элементов начинается с нуля.

6.7. Последовательность состоит из **различных** натуральных чисел и завершается числом 0. Определите значение второго по величине элемента в этой последовательности. Гарантируется, что в последовательности есть хотя бы два элемента.

6.8. Последовательность Фибоначчи определяется так:

*φ*0 = 0,  *φ*1 = 1,  *φn* = *φn*−1 + *φn*−2.

По данному числу *n* определите *n*-е число Фибоначчи *φn*.

Эту задачу можно решать и циклом for.

6.9. Дано натуральное число *A*. Определите, каким по счету числом Фибоначчи оно является, то есть выведите такое число *n*, что *φn = A*. Если *А* не является числом Фибоначчи, выведите число -1.

6.10. Дана последовательность натуральных чисел, завершающаяся числом 0. Определите, какое наибольшее число подряд идущих элементов этой последовательности равны друг другу.

### 7. Списки

Для хранения данных, заданных в виде последовательности, можно использовать структуру данных, называемую в Питоне список (в большинстве же языков программирования используется другой термин “массив”). Список представляет собой последовательность элементов, пронумерованных от 0, как символы в строке. Список можно задать перечислением элементов списка в квадратных скобках, например, список можно задать так:

*Primes = [2, 3, 5, 7, 11, 13]*

*Rainbow = ['Red', 'Orange', 'Yellow', 'Green', 'Blue', 'Indigo', 'Violet']*

Рассмотрим несколько способов создания и считывания списков. Прежде всего, можно создать пустой список (не содержащий элементов, длины 0), а в конец списка можно добавлять элементы при помощи метода append.

*a = [] # заводим пустой список*

*n = int(input()) # считываем количество элемент в списке*

*for i in range(n):*

*new\_element = int(input()) # считываем очередной элемент*

*a.append(new\_element) # добавляем его в список*

*# последние две строки можно было заменить одной:*

*# a.append(int(input()))*

*print(a)*

### Методы split и join

Элементы списка могут вводиться по одному в строке, в этом случае строку целиком можно считать функцией input(). После этого можно использовать метод строки split(), возвращающий список строк, которые получатся, если исходную строку разрезать на части по пробелам. Пример:

*s = input() # s == '1 2 3'*

*a = s.split() # a == ['1', '2', '3']*

Если хочется получить список именно из чисел, то можно затем элементы списка по одному преобразовать в числа:

*a = input().split()*

*for i in range(len(a)):*

*a[i] = int(a[i])*

Используя специальную магию Питона — генераторы — то же самое можно сделать в одну строку:

*a = [int(s) for s in input().split()]*

## Операции со списками

Со списками можно легко делать много разных операций.

|  |  |
| --- | --- |
| x in A | Проверить, содержится ли элемент в списке. Возвращает True или False |
| x not in A | То же самое, что not(x in A) |
| min(A) | Наименьший элемент списка |
| max(A) | Наибольший элемент списка |
| A.index(x) | Индекс первого вхождения элемента x в список, при его отсутствии генерирует исключение ValueError |
| A.count(x) | Количество вхождений элемента x в список |

Задачи

7.1. Выведите все элементы списка с четными индексами (то есть A[0], A[2], A[4], ...).

7.2. Выведите все четные элементы списка. При этом используйте цикл for, перебирающий элементы списка, а не их индексы!

7.3. Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

7.4. Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет — не выводите ничего. Если таких пар соседей несколько — выведите первую пару.

7.5. Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.

7.6. Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать.

Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200.

Выведите номер, под которым Петя должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.

7.7. Дан список, упорядоченный по неубыванию элементов в нем. Определите, сколько в нем различных элементов.

7.8. В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

7.9. Дан список чисел. Посчитайте, сколько в нем пар элементов, равных друг другу. Считается, что любые два элемента, равные друг другу образуют одну пару, которую необходимо посчитать.

7.10. Известно, что на доске 8×8 можно расставить 8 ферзей так, чтобы они не били друг друга. Вам дана расстановка 8 ферзей на доске, определите, есть ли среди них пара бьющих друг друга.

Программа получает на вход восемь пар чисел, каждое число от 1 до 8 — координаты 8 ферзей. Если ферзи не бьют друг друга, выведите слово NO, иначе выведите YES.

### 8. Функции и рекурсия

Функции — это такие участки кода, которые изолированы от остальной программы и выполняются только тогда, когда вызываются. Покажем, как написать функцию factorial(), которая принимает один параметр — число, и возвращает значение — факториал этого числа.

*def factorial(n):*

*res = 1*

*for i in range(1, n + 1):*

*res \*= i*

*return res*

*print(factorial(3))*

*print(factorial(5))*

Во-первых, код функции должен размещаться в начале программы, вернее, до того места, где мы захотим воспользоваться функцией factorial(). Первая строчка этого примера является описанием нашей функции. factorial — идентификатор, то есть имя нашей функции. После идентификатора в круглых скобках идет список параметров, которые получает наша функция. Список состоит из перечисленных через запятую идентификаторов параметров. В нашем случае список состоит из одной величины n. В конце строки ставится двоеточие.

Далее идет тело функции, оформленное в виде блока, то есть с отступом. Внутри функции вычисляется значение факториала числа n и оно сохраняется в переменной res. Функция завершается инструкцией return res, которая завершает работу функции и возвращает значение переменной res.

Инструкция return может встречаться в произвольном месте функции, ее исполнение завершает работу функции и возвращает указанное значение в место вызова. Если функция не возвращает значения, то инструкция return используется без возвращаемого значения. В функциях, которым не нужно возвращать значения, инструкция return может отсутствовать.

### Рекурсия *def short\_story():*

### *print("У попа была собака, он ее любил.")*

### *print("Она съела кусок мяса, он ее убил,")*

### *print("В землю закопал и надпись написал:")*

### *short\_story()*

Как мы видели выше, функция может вызывать другую функцию. Но функция также может вызывать и саму себя! Рассмотрим это на примере функции вычисления факториала. Хорошо известно, что 0!=1, 1!=1. А как вычислить величину n! для большого n? Если бы мы могли вычислить величину (n-1)!, то тогда мы легко вычислим n!, поскольку n!=n⋅(n-1)!. Но как вычислить (n-1)!? Если бы мы вычислили (n-2)!, то мы сможем вычисли и (n-1)!=(n-1)⋅(n-2)!. А как вычислить (n-2)!? Если бы... В конце концов, мы дойдем до величины 0!, которая равна 1. Таким образом, для вычисления факториала мы можем использовать значение факториала для меньшего числа. Это можно сделать и в программе на Питоне:

Задачи

8.1. Даны четыре действительных числа: *x1*, *y1*, *x2*, *y2*. Напишите функцию distance(x1, y1, x2, y2), вычисляющая расстояние между точкой *(x1,y1)* и *(x2,y2)*. Считайте четыре действительных числа и выведите результат работы этой функции.

8.2. Дано действительное положительное число *a* и **целоe** число *n*.

Вычислите *an*. Решение оформите в виде функции power(a, n).

Стандартной функцией возведения в степень пользоваться нельзя.

8.3. Напишите функцию capitalize(), которая принимает слово из маленьких латинских букв и возвращает его же, меняя первую букву на большую.

Например, print(capitalize('word')) должно печатать слово Word.

На вход подаётся строка, состоящая из слов, разделённых одним пробелом. Слова состоят из маленьких латинских букв. Напечатайте исходную строку, сделав так, чтобы каждое слово начиналось с большой буквы. При этом используйте вашу функцию capitalize().

Напомним, что в Питоне есть функция ord(), которая по символу возвращает его код [в таблице ASCII](http://ru.wikipedia.org/wiki/ASCII), и функция chr(), которая по коду символа возвращает сам символ. Например, ord('a') == 97, chr(97) == 'a'.

8.4. Дана последовательность целых чисел, заканчивающаяся числом 0. Выведите эту последовательность в обратном порядке.

При решении этой задачи нельзя пользоваться массивами и прочими динамическими структурами данных. Рекурсия вам поможет.

8.5. Напишите функцию fib(n), которая по данному целому неотрицательному *n* возвращает *n*-e число Фибоначчи. В этой задаче нельзя использовать циклы — используйте рекурсию.

Содержание

Введение …………………………………………………….3

Ввод и вывод данных .………………………………………4

Условия ………………………………………………………7

Вычисления ………………………………………………...11

Цикл for …………………………………………………….14

Строки ………………………………………………………16

Цикл while …………………………………………………..18

Списки ………………………………………………………19

Функции и рекурсия ……………………………………….22

Для заметок