

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 104, октябрь 2025 г.

Школа программирования

Основы программирования на языке Python¹

*Когда человек хочет передвинуть гору,
он начинает с того,
что убирает маленькие камни
Восточная мудрость*

Переменные величины. Ввод данных в программу. Инструкция присваивания

Переменные величины (или, короче, переменные) – величины, которые при выполнении программы могут принимать различные значения². Эти значения (в любой момент времени – какое-то одно) хранятся в памяти компьютера и могут быть использованы³, поэтому говорят, что переменные используются в программе для хранения информации.

Вспомним «математическую» программу из раздела 1. В ней ее пользователь должен ввести размеры сторон прямоугольника, которые могут быть любыми; эти значения будут использованы для расчета площади. Значение площади, которое тоже может быть любым, надо запомнить, а затем вывести на экран. Можем сказать, что в этой программе используются три переменные.

Каждая переменная характеризуется именем и типом.

В именах переменных можно использовать буквы, цифры (но имя не может начинаться с цифры) и знак подчеркивания «_». Желательно давать переменным «говорящие» имена, чтобы можно было сразу понять, зачем нужна та или иная переменная. Например, имя `discr` помогает понять, что эта переменная хранит значение дискриминанта квадратного уравнения.

Строчные и заглавные буквы различаются, то есть переменные с именами `dlna` и `Dlna` – это две разные переменные.

¹ См. также предыдущие выпуски интернет-журнала «Мир информатики».

² Существуют также так называемые «константы» – так в программировании называют величины, которые во время выполнения программы не меняются. В языке Python имеются также «встроенные» константы, например константы логического типа `True` и `False` (см. далее), константа `pi`, равная числу π (см. далее), и др.

³ О том, как используются значения переменных, вы, наверное, уже поняли из ранее опубликованных в журнале примеров их вывода на экран, – по имени (`print(x1)` и т. п.).

Есть набор слов, которые нельзя использовать в качестве имен переменных, так как эти слова «зарезервированы» в языке Python для определенных целей (эти слова называют «зарезервированными», или «служебными», или «ключевыми»).

Основные типы данных в языке Python приведены в таблице:

Тип	Обозначение типа	К нему относятся	Примеры	Примечание
Целый	<code>int</code>	Целые числа (положительные и отрицательные, а также 0)	4, -45, 0, 687	
Вещественный	<code>float</code>	Вещественные числа ⁴ (могут быть с дробной частью)	1.45, 0.00453, -3.789,	Как уже указывалось, в Python разделителем целой и дробной частей вещественного числа является точка
Логический	<code>bool</code>	Величины, которые могут принимать значения <code>True</code> («Истина») или <code>False</code> («Ложь»)		
Строковый	<code>str</code>	Последовательность (строка) символов, в том числе один символ или пустая строка ('')	'Школа', "красный", 'h', ''	Подробно о работе с переменными типа <code>str</code> будет рассказано в разделе 11

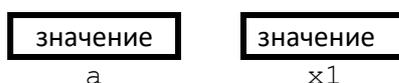
Тип величины определяет:

- какие значения может принимать величина (область допустимых значений переменной), например значением переменной типа `str` может быть только строка символов;

- какие операции можно проводить над переменной (множество допустимых операций с величиной); например, над величинами типа `float` можно проводить операции сложения, вычитания, умножения, деления и возведения в степень, а над величинами типа `int` – еще две операции (см. далее);

- какой объем памяти компьютера требуется для хранения значения данной переменной и в каком формате будут храниться данные. Например, переменные типа `float`, как правило, занимают 8 байтов.

Значение переменной хранится в каком-то месте памяти, которое можно смоделировать в виде прямоугольника (ячейки), рядом с которым указано имя этой переменной:



⁴ Их называют также «числами с плавающей точкой».

Как значение «падает» в соответствующий прямоугольник? Это происходит с помощью двух инструкций:

- 1) инструкции присваивания;
- 2) инструкции ввода данных.

Начнем со второй – инструкции⁵ `input()`. Она используется для ввода данных в программу в ходе ее выполнения с помощью клавиатуры (как говорят в программировании – «с клавиатуры»). Например, чтобы ввести значение переменной `a` строкового типа, нужно записать в программе:

```
a = input()
```

При выполнении этой строки на экране появится курсор, и система будет ожидать ввода значения с клавиатуры. Когда пользователь введет его и нажмет клавишу **<Enter>**, система запишет это значение в память в переменную `a`. До нажатия клавиши **<Enter>** можно удалять символы с помощью клавиши **<Backspace>**.

Когда приходится вводить значения нескольких величин, целесообразно указать в скобках сообщение-подсказку, чтобы пользователь программы знал, какое значение вводится в тот или иной момент ее выполнения:

```
fam = input('Введите фамилию: ')
im = input('Укажите имя: ')
```

Можно также вывести на экран общее сообщение:

```
print('Введите фамилию, а затем имя: ')
fam = input()
im = input()
```

Можно так оформить инструкции `print()` и `input()`, чтобы ввод значения на экране происходил в той же строке, где выведена подсказка:

```
print('Введите строку символов', end = '')
st = input()
```

Инструкция присваивания

Инструкция присваивания позволяет изменить (или задать впервые) значение переменной. В общем случае она оформляется так:

<имя переменной> = <выражение>

Например, когда присваивается значение величине числового типа (`int` или `float`), инструкция имеет вид:

<имя переменной> = <арифметическое выражение>

где *<арифметическое выражение>* – одно или несколько чисел, имен переменных величин или имен функций⁶, соединенных знаками⁷ арифметических операций.

Примеры:

```
c = 12
m = n
```

⁵Как и применительно к `print()`, мы используем термин «инструкция» (предписание, команда), так как это то, что должно быть выполнено программой. С другой стороны, это функция, так как в результате ее выполнения будет получено некоторое значение (строка символов). О том, что такое функция, см. дополнение в конце данного раздела.

⁶Также см. [дополнение](#).

⁷Их называют также «операторами».

```
sum = a + b
stepen3 = a ** 3
x = 2 * a - 3.6 * b/c;
sred = (a + b)/2
```

Обратим внимание на знаки возведения в степень («**») и деления («/»), а также на использование в последнем примере круглых скобок для изменения порядка действий (в Python, как и в других языках программирования, выражения записываются в строчку⁸, без «многоэтажных» дробей).

Распространенной ошибкой начинающих программистов является запись инструкций присваивания в виде:

```
c = 2a
```

(правильный вариант: `c = 2 * a`; вариант инструкции `c = a2` возможен, если `a2` – имя переменной, значение которой задано).

При определении порядка действий используется приоритет (старшинство) операций. Они выполняются в следующем порядке:

- действия в скобках;
- возведение в степень (**), справа налево (!);
- умножение (*) и деление (/), слева направо;
- сложение и вычитание, слева направо.

Таким образом, умножение и деление имеют одинаковый приоритет, более высокий, чем сложение и вычитание. Поэтому в последнем приведенном примере (`sred = (a + b) / 2`) запись без скобок привела бы к тому, что сначала выполнилось бы деление значения `b` на `2`, а затем сложение этого частного и значения переменной `a`.

Над величинами целого типа, кроме операций сложения, вычитания, умножения, деления и возведения в степень, можно выполнять также еще две операции:

1) определение целой части частного от деления одного целого числа на другое – знак операции «//»;

2) определение остатка от деления одного целого числа на другое – знак операции «%».

Пример программы:

```
a = 10
b = 3
z = a//b
m = a % b
print(z, m)
```

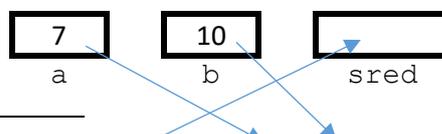
Результат её выполнения:

```
3 1
```

Как работает инструкция присваивания? Объясним на примере:

```
sred = (a + b)/2
```

Транслятор «посмотрит», что «лежит» в ячейках `a` и `b`, сложит соответствующие значения и сумму разделит на `2`. Результат «положится» в ячейку `sred`:



⁸ Такую запись называют «линейной».

```
sred = (a + b)/2
```

Это означает, что в программе может быть запись:

```
x = x + 10
```

невозможная в математике. Ее смысл в том, что нужно к имеющемуся в данный момент выполнения программы значению переменной x прибавить 10, а результат присвоить той же переменной.

В Python разрешено множественное присваивание. Запись

```
a, b, c = 7, 2, -5
```

равносильна инструкциям

```
a = 7
b = 2
c = -5
```

а запись $a = b = 0$ равносильна паре инструкций

```
b = 0
a = b
```

Часто используют сокращенную запись арифметических операций:

Сокращенная запись	Полная запись
$a += b$	$a = a + b$
$a -= b$	$a = a - b$
$a *= b$	$a = a * b$
$a /= b$	$a = a / b$

Рассмотрим задачу: «Даны два целых числа. Найти их сумму».

Ясно, что в программе следует использовать три переменные. Пусть их имена a , b , sum .

Программа решения задачи:

```
#Ввод исходных данных
print('Задайте первое число')
a = input()
print('Задайте второе число')
b = input()
#Расчет суммы
sum = a + b
#Вывод ответа на экран
print('Сумма этих чисел равна', sum)
```

Однако, после того как мы запустим программу и введем какие-то числа, допустим, 15 и 22, мы увидим странный ответ: 1522. Дело в том, что при нажатии клавиши на клавиатуре (в том числе клавиш с цифрами) в компьютер поступает код клавиши, то есть код соответствующего символа. И входные данные воспринимаются инструкцией `input()` именно как последовательность символов. Поэтому в приведенной программе переменные a и b будут заданы не как числа, а как цепочки символов, при сложении этих цепочек (с помощью оператора «+») программа просто объединяет их – приписывает вторую цепочку в конец первой.

Чтобы исправить эту ошибку, нужно преобразовать символьную строку, которая получена при вводе, в целое число. Это делается с помощью функции `int()` (от англ. *integer* – целый):

```
#Ввод исходных данных
print('Задайте первое число')
a = int(input())
print('Задайте второе число')
b = int(input())
...
```

Итак, после того как мы преобразовали введенные значения в формат целых чисел, программа работает правильно – складывает два числа, введенных с клавиатуры.

Если надо ввести в программу вещественное число, то при использовании инструкции `input()` необходимо записать функцию `float()`:

```
print('Задайте вещественное число')
z = float(input())
```

Как уже отмечалось, целесообразно включить в инструкцию `input()` подсказку:

```
a = int(input('Введите целое число '))
z = float(input('Задайте вещественное число '))
```

Мы рассмотрели особенности присваивания значения величинам числового типа.

Если нужно присвоить значение переменной типа `str`, то в правой части инструкции указывается строка символов в кавычках (одинарных или двойных), имя переменной величины типа `str` или имя метода для работы со строками. Примеры:

```
club = 'Спартак'
moi_club = club

im = 'Dima'
im2 = im.upper() #0 методе upper() см. раздел 3
```

Можно также указать несколько таких значений, между которыми записывается знак «+»:

```
Kto = im + fam
all = 'Я - болельщик команды' + club
```

Другие операции⁹ над строками выполнять нельзя (помните, что перечень операций определяется типом величины?).

Инструкция присваивания значений величинам типа `bool` будет рассмотрена в одной из будущих публикаций.

Обсудим одну интересную задачу программирования: «Даны значения двух переменных величин `a` и `b`. Произвести обмен их значений».

Кажущее очевидным решение в виде:

```
a = b
b = a
```

или

```
b = a
```

⁹ Возможна также операция «умножения» величины строкового типа на целое число, которая используется редко.

```
a = b
```

требуемого результата не даст (убедитесь в этом!). Как же быть? А так, как происходит обмен содержимого двух чашек, в одной из которых находится молоко, а в другой – чай. Нужна третья чашка!¹⁰ То есть в нашей задаче для решения требуется третья, вспомогательная переменная. С ее использованием обмен может быть проведен следующим образом:

```
c = a      #Запоминаем значение величины a
a = b      #Величине a присваиваем значение величины b
b = c      #Величине b присваиваем 'старое' значение величины a
```

или

```
c = b
b = a
a = c
```

Так задача решается во всех языках программирования. В программе на Python имеется также возможность провести обмен оригинальным¹¹ способом:

```
a, b = b, a
```

Здесь использовано множественное присваивание (см. выше).

Контрольные вопросы

1. Какие величины в программе называют «переменными»?
2. Чем характеризуется каждая переменная?
3. Каковы правила присвоения имен переменным?
4. Почему желательно переменным давать «говорящие» имена?
5. Какие типы переменных вы знаете?
6. Что определяет тип переменной?
7. Какие значения может иметь переменная логического типа?
8. Как можно смоделировать хранение значения переменной в памяти компьютера?
9. Как обратиться к значению (использовать значение) переменной величины в программе?
10. С помощью какой инструкции можно ввести в программу значение переменной в ходе ее выполнения?
11. В чем заключается особенность ввода в программу в ходе ее выполнения числовых значений переменных?
12. Почему желательно выводить на экран подсказку перед вводом данных?
13. С помощью какой инструкции можно изменить (или задать впервые) в программе значение переменной? Как она оформляется в общем случае? Как она оформляется применительно к переменным числового типа? К переменным типа `str`?
14. Что такое приоритет операций? Зачем он нужен? Перечислите арифметические операции в порядке уменьшения приоритета.

¹⁰ Можно, конечно, использовать и две дополнительные чашки, но это уже, так сказать, «слишком» ☺.

¹¹ Далее будут приведены и другие оригинальные методы и функции Python. Но наряду с ними будут обсуждаться способы решения соответствующих задач без использования возможностей Python. Это полезно, потому что они заставляют думать, как решить соответствующую задачу, и раскрывают суть того, как ее решает система программирования. Кроме того, эти способы решения могут понадобиться вам, когда вы будете программировать на других языках.

15. В каком порядке выполняются операции, если они имеют одинаковый приоритет?

16. Зачем в инструкции присваивания используются скобки?

17. Чем отличаются операции, знаки которых «/», «//» и «%»?

Задания

1. Запишите в одну строку по правилам языка Python следующие арифметические выражения:

а) $\frac{-1}{x^2}$;

б) $\frac{a}{bc}$;

в) $\frac{a}{b}c$;

г) $\frac{(a+b)}{2}$

д) $5,45 \frac{(a+2b)}{2-b}$;

е) $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$;

ж) $\frac{-b + \frac{1}{a}}{\frac{2}{c}}$;

з) $\frac{1}{1 + \frac{a+b}{2}}$;

и) $1 + \frac{1}{2 + \frac{1}{2 + \frac{3}{5}}}$;

й) 2^{m^n} .

2. Запишите следующие выражения (представленные в линейной записи) в обычную форму: а) $a/b/c$;

б) $a+b/c$;

в) $(a+b)/c$;

г) $a*b/(c+2)$;

д) $a/b*(c-3)/d$;

е) $(a/3+b/2)/(b + c/(2+b))$

Задания 3–7, приведенные ниже, удобно выполнять с помощью таблицы, имитирующей «трассировку»¹² программы. Например, если требуется определить значение переменной y после выполнения следующего фрагмента программы:

$x = 15$

$x = x + 6$

$y = -x + 26$

$y = y - x$

то такая таблица имеет вид:

Инструкция	x	y
$x = 15$	15	
$x = x + 6$	$15 + 6 = 21$	
$y = -x + 26$	21	$-21 + 26 = 5$
$y = y - x$	21	$5 - 21 = -16$

Ответ: -16.

¹² Трассировка – это процесс выполнения программы по шагам, инструкция за инструкцией, с остановками после каждого шага.

3. Определите значение переменной c после выполнения следующего фрагмента программы:

```
a = 5
a = a + 6
b = -a
c = a - 2 * b
```

4. Определите значение переменной v после выполнения следующего фрагмента программы:

```
a = 40
b = 10
b = -a/2 * b
v = b + a * 2
```

5. Определите значения переменных s и k после выполнения следующего фрагмента программы:

```
s = 14
k = -3
d = s + 1
s = d
k = 2 * s
```

6. Определите значения переменных p и q после выполнения следующего фрагмента программы:

```
p = 0
q = 30
d = q - 5
q = 2 * d
p = q - 100
```

7. Вычислите значение переменной z при $x = 25$ и $y = 7$:

а)
 $z = x \% y + y$

б)
 $z = x // y + x$

в)
 $y = x // y$
 $z = x // y$

г)
 $y = x // y + y$
 $z = x \% y + x$

д)
 $y = x \% y + 4$
 $z = x \% y + 1$

е)
 $y = x // y$
 $z = x \% (y + 1)$

ж)
 $y = x \% y$
 $z = x // (y + 1)$

Задания на разработку программ

1. Напишите программу, в которую вводится имя человека и выводится на экран приветствие в виде слова 'Привет!', после которого должна стоять запятая, введенное имя и восклицательный знак. После запятой должен стоять пробел, а перед восклицательным знаком пробела быть не должно.

2. Напишите программу, в которую вводится целое число, после чего на экран выводится следующее и предыдущее целое число. Например, при вводе числа 15 на экран должно быть выведено:

Следующее за числом 15 число - 16.

Для числа 15 предыдущее число - 14.

3. Напишите программу, в которой вычисляются сумма, разность, произведение, частное и среднее арифметическое двух целых чисел, введенных с клавиатуры. Например, при вводе чисел 2 и 7 должен быть получен ответ вида:

```
2+7=9 2-7=-5 2*7=14 2/7=0.2857142857142857 (2+7)/2=4.5
```

4. Напишите программу, которая решает следующую задачу: «N школьников делят k яблок поровну так, чтобы каждому достались только целые яблоки, остальные яблоки остаются в корзинке. Определить, сколько яблок достанется каждому школьнику и сколько яблок останется в корзинке».

5. Напишите программу, в которой рассчитывается сумма цифр двузначного числа, вводимого с клавиатуры.

6. Напишите программу, в которую вводится трехзначное число и выводятся на экран его цифры. Например, при вводе числа 123 программа должна вывести:

```
1, 2, 3
```

7. Напишите программу, в которой рассчитывается:

- а) сумма цифр 4-значного числа, вводимого с клавиатуры;
- б) то же, 5-значного.

Дополнение

Возникает вопрос: «А можно ли в программах на Python извлекать квадратный корень?» - Можно. Для этого следует использовать функцию `sqrt()`. В программировании функция – «самостоятельная» программа, имеющая имя и решающая какую-то частную, вспомогательную задачу. Функции бывают:

1) стандартные («встроенные» в систему программирования и доступные без всяких условий). Например, стандартными являются уже упоминавшиеся ранее функции `print()` (решает задачу вывода информации на экран), `input()` (возвращает в программу строку символов, введенных с клавиатуры), `int()` (преобразует строку символов-цифр в число¹³), `float()` и др.;

2) нестандартные – созданные пользователем. Подробно о таких функциях рассказано в разделе 16;

3) полустандартные¹⁴ – предусмотренные в системе программирования, но для вызова (использования) которых необходимо подключить к разрабатываемой программе модуль¹⁵, в который входят такие функции.

Функция `sqrt()` относится к последней группе. Для ее применения в программе следует подключить модуль `math`. Для этого в начале программы¹⁶ нужно записать:

```
import math
```

Это инструкция импорта (загрузки) модуля.

¹³ Если в качестве аргумента функции `int()` указано вещественное число, то функция вернет целую часть числа.

¹⁴ Термин предложен автором.

¹⁵ Модуль – элемент системы программирования, включающий в себя группу «полустандартных» функций.

¹⁶ Если быть точным – то до первого использования функции.

Сама функция вызывается для выполнения, как вы уже поняли из приведенных ранее примеров, по ее имени. Для функции `sqrt()` это делается так:

```
kor = math.sqrt(...) #Используется точечная запись
```

где вместо ... указывается арифметическое выражение (см. выше):

```
kor2 = math.sqrt((a + b)/2)
print(math.sqrt(n))
if math.sqrt(x) > 2:
...
```

и т. п.

Обратим внимание на то, что функция `sqrt()` возвращает вещественный результат даже в случаях `sqrt(1)`, `sqrt(4)`, `sqrt(49)` и т. п.

Кроме `sqrt()`, в модуле `math` имеется еще ряд математических функций для работы с вещественными числами:

`sin(x)`, `cos(x)`, `tan(x)` – определяют, соответственно, синус, косинус и тангенс x (значение x указывается в радианах);

`log10(x)` и `log(x)` – возвращают, соответственно, десятичный и натуральный логарифмы x

и другие.

Модуль также определяет две константы:

`pi` – число π ;

`e` – число e (основание натурального логарифма).

При импорте модуля в виде

```
import math
```

в программе будут доступны все его функции и константы.

Можно подключить только нужные функции:

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

В этом случае все функции модуля `math`, кроме перечисленных (`sqrt()`, `sin()`, `cos()`), в программе будут недоступны. Перечисленные при импорте конкретные функции можно вызывать без использования точечной записи:

```
r = sqrt((a + b)/2)
print(sin(n))
if sqrt(x) > 2:
```

Контрольные вопросы

1. Что такое «функция»?
2. Какие виды функций возможны в программах на Python? В чем особенность каждого вида?

Задачи для разработки программ

1. Даны катеты прямоугольного треугольника. Определить его гипотенузу.
2. Даны координаты на плоскости двух точек. Найти расстояние между этими точками.

Если вы, читатель, хотите стать программистом, то имейте в виду, что одним из условий этого является большой опыт решения задач по программированию. Решайте задачи, абстрактные и содержательные, на 5 минут и на день работы. Все это обязательно вам пригодится. Чем больше опыт решения задач, тем больше вероятность, что уже решенная задача или аналогичная ей встретится вам в будущем.

Найти два натуральных числа

В статье описана методика разработки программы решения следующей задачи «Найти два натуральных числа, меньших 100, таких, что сумма их квадратных корней минимально отличается от ближайшего целого числа. Искомые числа не должны быть полными квадратами». Как принято в нашем журнале, используем школьный алгоритмический язык (система программирования КуМир). Русский синтаксис этого языка делает приводимые методики и программы максимально понятными, и вы сможете разработать аналогичную программу на языке программирования, который изучаете.

Прежде всего, предлагаем читателям подумать над тем, почему в условии есть уточнение о полных квадратах?

Теперь – о программе.

В ней потребуются решение двух частных, вспомогательных задач:

- 1) нахождение ближайшего к вещественному (сумме квадратных корней чисел) целого числа;
- 2) нахождение минимального числа из некоторого набора чисел.

Для решение первой задачи, надо вспомнить правила округления дробных чисел: если число имеет дробную часть меньше 0,5, то ближайшее целое число будет равно целой части данного числа, иначе – на 1 больше целой части.

Алгоритм нахождения минимального числа из некоторого набора чисел аналогичен алгоритму действий человека, который определяет максимальное значение в некотором наборе:

14, 6, 9, 7, 3, 16, 10, 2, 7

– сначала он запоминает первое число, а затем рассматривает второе число. Если оно меньше того числа, которое помнил, то запоминает новое число и переходит к следующему, третьему числу; в противном случае просто переходит к следующему числу и делает то же самое.

Так и в программе сначала следует задать первое число и принять его в качестве минимального *мин*. Затем задаются остальные числа, и каждое из них сравнивается со значением *мин*. Если очередное число меньше *мин*, то оно принимается в качестве нового значения *мин*.

Если диапазон обрабатываемых чисел известен, то можно отдельно не вводить и запоминать первое число, а в качестве начального значения переменной *мин* принять число, заведомо большее всех обрабатываемых чисел. В нашей задаче в качестве такого начального значения можно принять число 0,5.

В программе используем следующие переменные величины:

n1, n2 – обрабатываемые числа;

сумма_корней – сумма их квадратных корней;

ближ_целое – целое число, ближайшее к вещественному;

разность – разность значений *сумма_корней* и *ближ_целое*;

мин_разность – минимальная разность.

Обсудим ещё два вопроса.

1. Как проверить, является ли целое число полным квадратом?

Ответ такой – если квадратный корень из этого числа равен своей целой части; для числа 25,0 ответ будет положительным (корень из 25 равен 5,0), для числа 26,0 – отрицательным (корень из 25 равен 5,0999).

2. Как организовать перебор все пар чисел, меньших 100?

Это можно сделать с помощью вложенного оператора цикла:

```
нц для n1 от 2 до 99
  нц для n2 от 2 до 99
```

Но при этом:

- каждая пара чисел будет рассматриваться дважды (например, 13 с 67 и 67 с 13);
- каждое число будет сравниваться с самим собой.

В данном случае (диапазон обрабатываемых чисел небольшой) это увеличит время выполнения программы ненамного. В заданиях ЕГЭ по информатике, в которых этот диапазон может быть гораздо больше, лучше использовать такой вариант вложенного цикла:

```
нц для n1 от 2 до 98
  нц для n2 от n1 + 1 до 99
```

Проанализируйте его и убедитесь, что каждая пара чисел рассматривается только один раз).

Итак, вся программа:

```
алг
нач цел n1, n2
  вещ сумма_корней, ближ_целое, разность, мин_разность
  мин_разность := 0.5 | Начальное значение
  нц для n1 от 2 до 98
    нц для n2 от n1 + 1 до 99
      | Проверяем, не являются ли числа полными квадратами
      если sqrt(n1) <> int(sqrt(n1)) и sqrt(n2) <> int(sqrt(n2))
        | Только в этом случае делаем расчёты
        то
          сумма_корней := sqrt(n1) + sqrt(n2)
          | Находим ближайшее целое число
          если сумма_корней - int(сумма_корней) < 0.5
            то
              ближ_целое := int(сумма_корней)
            иначе
              ближ_целое := int(сумма_корней) + 1
          все
          разность := abs(сумма_корней - ближ_целое)
          если разность < мин_разность
            то
              мин_разность := разность
          все
        все
      кц
    кц
  | Вывод минимальной разности
  вывод мин_разность
кон
```

Решили ли мы поставленную задачу? – Нет, не решили. Мы определили только минимальное отклонение суммы квадратных корней от ближайшего целого числа, а нужно было установить для каких двух чисел это имеет место.

Для нахождения этих чисел следует использовать ещё две переменные величины — $nn1$ и $nn2$, которые будут хранить (см. предыдущую статью) информацию о двух числах, для которых указанное отклонение является минимальным в ходе рассмотрения всех пар чисел:

```
если разность < мин_разность
то
  | Запоминаем не только новую минимальную разность
  мин_разность := разность
  | но и числа, при которых она получается
  nn1 := n1
  nn2 := n2
все
```

Разработайте программу (на языке программирования, который изучаете) и определите искомые числа. Результаты присылайте в редакцию.

Задачник

Задача, которую вы решаете, может быть очень скромной, но если она бросает вызов вашей любознательности и если вы решите её собственными силами, то вы сможете испытать ведущее к открытию напряжение ума и насладиться радостью победы.

Джордж Поля

Кто кем работает?

В семье пять человек: муж, жена, их сын, сестра мужа и отец жены. Все они работают. Один — инженер, другой — юрист, третий — слесарь, четвертый — экономист, пятый — учитель информатики.

Известно, что:

- 1) юрист и учитель не являются кровными родственниками;
 - 2) слесарь — хороший спортсмен. Он пошел по стопам экономиста и играет в футбол за сборную своего завода. Они оба мужчины;
 - 3) инженер старше жены своего брата, но моложе, чем учитель.
- Определите профессию каждого члена семьи.

Четыре девочки

На улице, встав в кружок, беседуют четыре девочки: Аня, Валя, Надя и Галя. Известно, что:

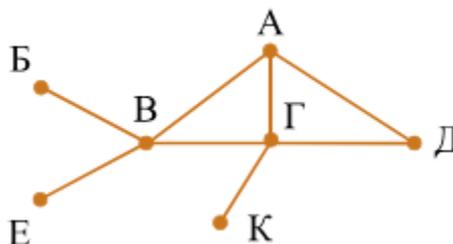
- 1) девочка в зеленом платье — не Аня и не Валя — стоит между девочкой в голубом платье и Надей;
 - 2) девочка в белом платье стоит между девочкой в розовом платье и Валей.
- Можно ли определить, какого цвета платье у каждой из девочек?

Три задачи из ЕГЭ по информатике

Задача 1

На рисунке справа схема дорог Н-ского района изображена в виде графа; а в таблице слева содержатся сведения о протяжённости каждой из этих дорог (в километрах).

	п1	п2	п3	п4	п5	п6	п7
п1		7					
п2	7		8		3	4	
п3		8		11	6		
п4			11		5		
п5		3	6	5			9
п6		4					
п7					9		



Так как таблицу и схему рисовали независимо друг от друга, то нумерация населённых пунктов в таблице никак не связана с буквенными обозначениями на графе. Определите, какова длина дороги из пункта А в пункт Г. В ответе запишите целое число — так, как оно указано в таблице.

Задача 2

Миша заполнял таблицу истинности логической функции F :

$$(x \wedge \neg y) \vee (x \equiv z) \vee w,$$

но успел заполнить лишь фрагмент из трёх различных её строк, даже не указав, какому столбцу таблицы соответствует каждая из переменных w, x, y, z .

				F
		0	1	0
1	0		1	0
1	1	0		0

Определите, какому столбцу таблицы соответствует каждая из переменных w, x, y, z в том в ответе напишите буквы w, x, y, z в том порядке, в котором идут соответствующие им столбцы (сначала буква, соответствующая первому столбцу; затем буква, соответствующая второму столбцу, и т. д.). Буквы в ответе пишите подряд, никаких разделителей между буквами ставить не нужно.

Пример. Функция F задана выражением $\neg x \vee y$, зависящее от двух переменных, а фрагмент таблицы имеет следующий вид:

		F
0	1	0

В этом случае первому столбцу соответствует переменная y , а второму столбцу — переменная x . В ответе следует написать: yx .

Задача 3

По каналу связи передаются сообщения, содержащие только семь букв: А, Б, В, Г, Й, К, Л. Для передачи используется двоичный код, удовлетворяющий условию Фано. Кодовые слова для некоторых букв известны: Б — 00, Г — 010, К — 101. Какое наименьшее количество двоичных знаков потребуется для кодирования слова БАЛАЛАЙКА?

Примечание. Условие Фано означает, что ни одно кодовое слово не является началом другого кодового слова.

Максимальная сумма разностей

Вот задача: «Расставить в таблице числа от 1 до 16 так, чтобы сумма абсолютных значений разностей чисел в смежных клетках таблицы была максимально возможной. Всего должны быть учтены 24 разности (12 – по горизонтали и 12 – по вертикали)».

Так как общее число различных вариантов размещения 16 чисел очень большое (оно равно $1 \times 2 \times \dots \times 16$, или так называемому «факториалу числа 16, и составляет $2,09228 \times 10^{13}$), то решение задачи перебором вариантов невозможно. Попробуйте решить её на основе рассуждений и используя программу Microsoft Excel (или аналогичную программу – электронную таблицу).

Так как в условии идёт речь о максимально возможной сумме разностей, то это значит, что небольшие числа должны чередоваться с большими. Так определилось место чисел 1 и 2, например:

	1		
		2	

		1	
	2		

«Вокруг» этих чисел следует расставить большие числа, а затем и остальные так, чтобы указанная в условии сумма разностей была максимальной. При такой расстановке все расчёты целесообразно проводить в среде электронной таблицы.

Оформите верхнюю часть листа программы следующим образом:

	A	B	C	D	E	F	G	H
1								
2								
3								
4								
5								
6								
7								
8								
9								

В диапазоне, выделенном зелёным цветом, записывайте расставляемые числа, оформленном жёлтым цветом – рассчитывайте абсолютные значения разностей смежных чисел по горизонтали, синим – то же, по вертикали.

В программе Microsoft Excel абсолютное значение выражения возвращает функция ABS. Следовательно, заполнить жёлтый диапазон можно, введя в ячейку F1 формулу =ABS(B1-A1), которую затем распространить (скопировать) на остальные ячейки этого диапазона.

Аналогично заполняется синий диапазон.

Сумму всех абсолютных значений разностей смежных чисел рассчитывайте в ячейке F8, используя две функции СУММ. Напомним, что эта сумма должна быть максимальной.

Ответ (таблицу с размещением чисел) и соответствующую сумму разностей присылайте в редакцию. Все приславшие правильный ответ будут награждены дипломами.

Криптография

Частотный анализ

В нашем журнале описаны несколько методов шифрования текста, которые объединяет одна особенность — все они были основаны на замене исходных букв другими символами [1–5]. Восстановить буквы текста, зашифрованного путем замены, с большой уверенностью можно, анализируя частоту появления тех или иных букв и их сочетаний. Этот метод (он так и называется — частотный анализ) основывается на том, что известно, как часто встречается та или иная буква в русском языке (или в английском языке — именно это учитывал герой рассказа Эдгара По “Золотой жук”, расшифровывая найденный пергамент). Даже если в каких-то частях текста возникает неоднозначность, она легко устраняется по смыслу.

Относительные частоты букв русского языка указаны в табл. 1.

Таблица 1

Буква	Относительная частота	Буква	Относительная частота	Буква	Относительная частота
а	0,062	к	0,028	ф	0,002
б	0,014	л	0,035	х	0,009
в	0,038	м	0,026	ц	0,004
г	0,013	н	0,053	ч	0,012
д	0,025	о	0,090	ш	0,006
е, ё	0,072	п	0,023	щ	0,003
ж	0,007	р	0,040	ы	0,016
з	0,016	с	0,045	ь, ъ	0,014
и	0,062	т	0,053	э	0,003
и	0,010	у	0,021	ю	0,006
				я	0,018

Буквы *е* и *ё*, а также *ь*, *ъ* кодируются обычно одинаково, поэтому в таблице они не различаются. Как следует из таблицы, наиболее частая буква русского языка — *о*. Ее относительная частота, равная 0,090, означает, что на 1000 букв русского текста приходится в среднем 90 букв *о*. В таком же смысле понимаются относительные частоты и остальных букв. В табл. не указан еще один “символ” — промежуток между словами (пробел). Его относительная частота наибольшая и равна 0,175.

Задание для самостоятельной работы

Используя табл. 1, расшифруйте такую криптограмму:

Цярснсмци ямякзж онкдждм мд снкыйн гкю онгрсямнбнцмщф
йпзоснвпялл мн б гптвзф рктцяюф нм ркнемдд.

Ответы, пожалуйста, присылайте в редакцию.

Десять ребусов на одну тему

Решите ряд ребусов.

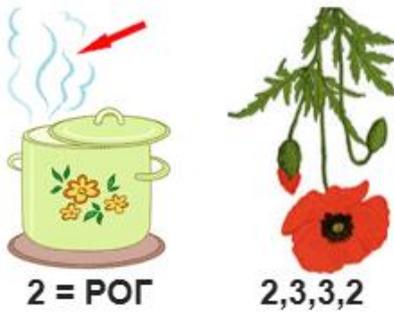
Ребус 1



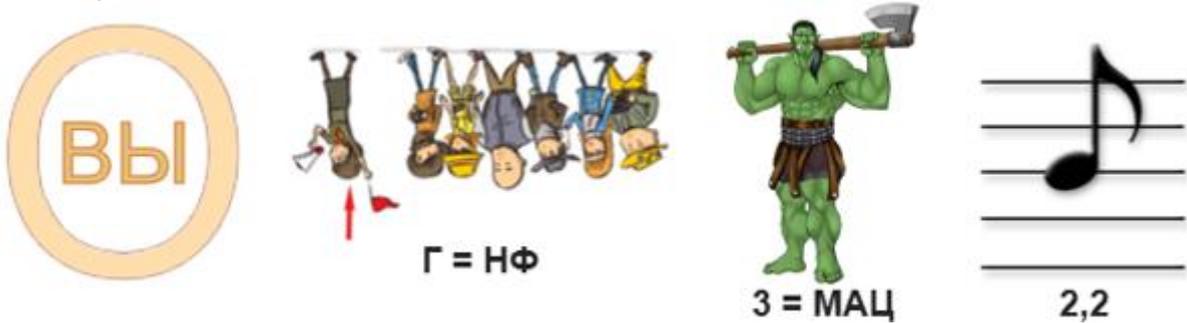
Ребус 2



Ребус 3



Ребус 4



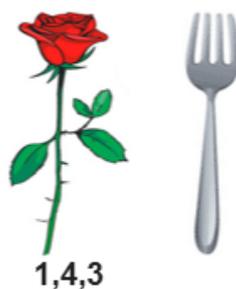
Ребус 5



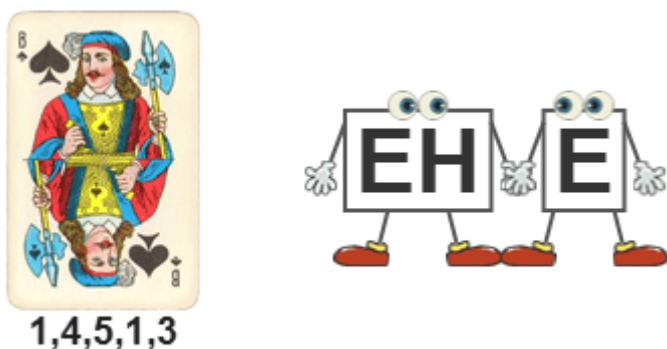
Ребус 6



Ребус 7



Ребус 8



Ребус 9



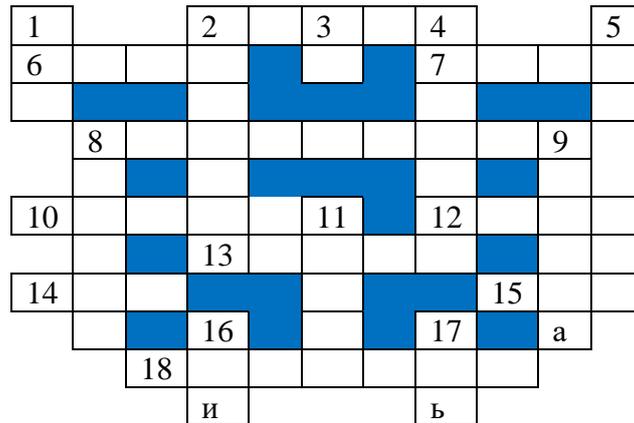
Ребус 10



Установите также, с какой темой школьного курса информатики они связаны. Ответы (можно не на все ребусы) присылайте в редакцию.

Кроссворд

Решите, пожалуйста, кроссворд



По горизонтали

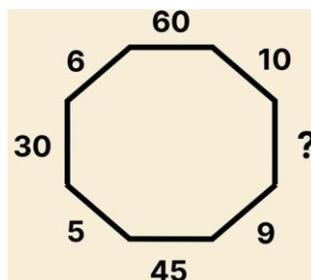
2. Поименованная группа файлов, объединенных по какому-то признаку.
6. Система знаков для секретного письма.
7. Французский инженер и изобретатель системы кодировки символов для телетайпов, в честь которого названа единица скорости передачи информации.
8. Постоянная величина в программе.
10. Знак препинания.
12. Русский вариант названия операционной системы для советских персональных компьютеров БК-0010, а также северный ветер.
13. Машина с человекоподобными действиями, управляемая по программе.
14. Нижняя граница стека.
15. Буква греческого алфавита, которой, как правило, обозначают неизвестную величину.

По вертикали

1. Парный орган — приемник информации у человека.
2. Нестандартное устройство для вывода информации в компьютере.
3. 3,1415926.
4. Пользователь телефонной линии.
5. Число в системе условных обозначений символов.
8. Почти антоним к слову “клиент”.
9. Часть света, в которой будут выпускаться компьютеры “черной сборки”.
11. В текстовом редакторе Microsoft Word — текст, набранный до нажатия клавиши <Enter>.
16. Цифра шестнадцатеричной системы счисления.
17. ... координат.

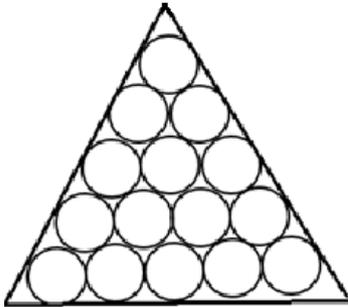
Число вместо вопросительного знака

Какое число должно стоять вместо вопросительного знака?



15 окружностей

В равностороннем треугольнике вписаны 15 окружностей. Закрасьте их тремя цветами – красным, зелёным и жёлтым так, чтобы расстояние между каждой парой покрашенных кругов одного цвета было одним и тем же. Круги одного цвета не должны касаться друг друга.



В ответе можно круги не закрашивать, а обозначить их буквами *к*, *з* и *ж*.

Магические квадраты

Магический (или волшебный) квадрат – это квадратная таблица, заполненная числами без их повторений таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова. Эту сумму называют «магической суммой».

На рис. 1–3 показаны части магических квадратов размером 3×3 . Получите квадраты полностью, заполнив свободные клетки.

4		
	6	7

Рис. 1

3		6
	2	

Рис. 2

	1	
	25	4

Рис. 3

Подсказка

Найдите в Интернете информацию о том, как магическая сумма связана с числом в центре магического квадрата.

Для решения удобно использовать электронную таблицу Microsoft Excel или аналогичную программу.

В одной из ячеек введите то или иное число, а в оттенённые ячейки запишите формулы для расчёта сумм в строках, столбцах и на диагонали. Эти формулы помогут контролировать правильность заполнения квадрата.

	A	B	C	D	E
1					
2	3		6		
3					
4		7			
5					
6					

Числа и степени. Два вопроса

1. Есть только одна пара целых чисел p и q , для которых $p^q = q^p$. Какие это числа?
2. При каких целых значениях p , q и r справедливо равенство $(p^q)^r = p$ в степени q^r ?

Преобразовать таблицу

Дана таблица с числами (см. рис. 1 слева). Необходимо разместить числа так, как показано на рис. 1 справа, по следующим правилам – можно менять местами два числа, клетки которых отстоят друг от друга ходом шахматного коня. Например (см. рис. 2), число 6 можно поменять местами с числами 4, 8, 13 или 15.

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16



16	12	8	4
15	11	7	3
14	10	6	2
13	9	5	1

Рис. 1

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Рис. 2

Четыре числа

В таблице записаны цифры, которые при их прочтении слева направо и сверху вниз образуют четыре числа, сумма которых равна 67. Какие это числа? Нулей в таблице нет.

?	?
?	?

Необычный шахматный слон

В нижнем ряду фрагмента шахматной доски (см. рисунок ниже) стоит шахматная фигура, которую назовём «полуслон», так как она, в отличие от настоящего шахматного слона, ходит на любое количество клеток по диагонали только влево вверх или вправо вверх. Сколько существует способов перемещения фигуры до верхнего ряда данного фрагмента?

— Назарян Армен, Москва, школа № 1530 «Школа Ломоносова», учитель
Каменецкая Т.С.;
— Пусь Иван, г. Пенза, школа № 73, учитель **Диков А.В.**

Спасибо всем!