

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 103, сентябрь 2025 г.

Школа программирования

Основы программирования на языке Python

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

Редакция продолжает публикацию фрагментов книги с одноимённым названием,
изданной издательством «ДМК Пресс»
<https://dmkpress.com/catalog/computer/programming/python/978-5-97060-641-4/> См. также:

- понятия «алгоритм» и «программа» // Мир информатики, выпуск 102, август 2025 г.;
- Python. Варианты действий в программе // Мир информатики, выпуск 90, июнь 2024 г.;
- Python. Повторение действий в программе // Мир информатики, выпуск 90, июнь 2024 г.; Мир информатики, выпуск 91, август 2024 г.;

Вывод информации на экран

Для вывода информации на экран в программах на языке Python используется инструкция `print()`. В скобках указывается то, что нужно вывести. Например, чтобы вывести на экран приветствие, надо записать:

```
print('Привет!')
```

или

```
print("Привет!")
```

то есть текст (последовательность символов) указывается в кавычках.

В результате выполнения инструкции на экран будут выведены все символы, указанные в кавычках, включая начальные и конечные пробелы.

Можно также указывать в скобках:

- число:

```
print(5)  
print(-2)  
print(3.14)
```
- имя переменной величины¹:

```
print(a)  
print(x1)
```

¹ О переменных будет рассказано в следующей публикации.

```
print(perimetr)
```

- арифметическое или логическое выражение (о них будет рассказано ниже).

В программах на языке Python могут использоваться также объекты, над которыми выполняются некоторые действия (методы). Например, к последовательности символов может быть применен метод `upper()`, преобразующий все буквы в их написание в верхнем регистре («прописными буквами»). Чтобы эти действия были выполнены (был применен метод), следует записать так:

```
<ИМЯ ОБЪЕКТА>.<ИМЯ МЕТОДА>
```

Например:

```
famil.upper()
```

Такую запись (которую называют «точечной») также можно использовать для вывода на экран:

```
print(famil.upper())
```

Результат выполнения программы для каждого из указанных случаев записи инструкции показан в таблице:

Что указано в скобках	Пример	На экран будет выведено	Пример
1. Текст	<code>print('Привет!')</code>	Текст без кавычек, включая возможные начальные и конечные пробелы	Привет!
2. Число	<code>print(-2)</code>	Соответствующее число	-2
3. Имя переменной величины	<code>print(x1)</code>	Значение величины	273
4. Выражение	<code>print(a * b)</code>	Значение выражения	1024
5. Метод	<code>print(famil.upper())</code>	Результат применения метода	ЛУКИН

Можно указывать несколько значений, в том числе разного типа, разделяя их запятой. Например, в программе решения задачи расчета площади и периметра любого прямоугольника может быть использован следующий вариант инструкции:

```
print(1, '. Площадь прямоугольника равна', pl, 'кв. см')
```

в котором указаны число, два текста и имя переменной величины. В результате выполнения программы на экран будет выведено примерно следующее:

```
1. Площадь прямоугольника равна 42 кв. см
```

Видно, что между указанными в инструкции `print()` значениями (будем называть их «список вывода») выводится также один пробел. Этот разделитель можно изменить на любой другой символ (или последовательность символов). Для этого после списка вывода нужный символ-разделитель указывается как параметр `sep` инструкции, например:

```
print(<СПИСОК ВЫВОДА>, sep = ', ')
```

Можно использовать в качестве разделителя «пустой» символ (' '):

Пример	На экран будет выведено
<pre>print(a, '+', b, '=', c, sep = ' ')</pre> #a и b – заданные числовые значения #c – их сумма	2+3=5

Во всех перечисленных случаях каждая новая инструкция `print()` выводит соответствующие значения на следующей строке. Чтобы исключить это, необходимо указать другой параметр этой инструкции – `end`, задав его равным «пустому» символу:

```
print(<СПИСОК ВЫВОДА>, end = '')
```

Конечно, можно указывать и оба параметра:

```
print(<СПИСОК ВЫВОДА>, sep = ', ', end = '')
```

или

```
print(<СПИСОК ВЫВОДА>, end = '', sep = ', ')
```

Для вывода пустых строк следует использовать инструкцию `print()` без списка вывода:

Пример	На экран будет выведено
<pre>print('1.')</pre>	1.
<pre>print()</pre>	
<pre>print('2.')</pre>	2.

Вывод на экран вещественных чисел (могут быть с дробной частью) имеет особенности. В результате выполнения инструкций

```
print(1/3)
```

и

```
a = 1
b = 3
print(a/b)
```

на экран будет выведено следующее:

```
0.3333333333333333
```

Дело в том, что операция деления (знак операции «/») дает вещественный результат даже в случаях, когда она проводится над целыми числами, в том числе в случаях 25/5, 48/12 и т. п.

Количество цифр в дробной части можно ограничить. Для этого в инструкции `print()` перед выводимым значением следует записать точку, количество цифр КолДробн в дробной части и букву `f` в виде:

```
print('% .3f ' % <значение>)2
```

или

```
print("%.2f " % <значение>)3
```

где <значение> – выводимое значение.

² Между первыми двумя символами ' % пробела быть не должно. В противном случае он также будет выведен.

³ Между первыми символами "% пробела быть не должно.

Примеры:

Программа	На экран будет выведено														
<pre>a = 234.193 print('% .1f ' % a)</pre>	234.2 <table><tr><td></td><td>2</td><td>3</td><td>4</td><td>.</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table> Позиция экрана слева		2	3	4	.	2	1	2	3	4	5	6		
	2	3	4	.	2										
1	2	3	4	5	6										
<pre>a = -234.123 print('% .2f ' % a)</pre>	-234.12 <table><tr><td>-</td><td>2</td><td>3</td><td>4</td><td>.</td><td>1</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table> Позиция экрана слева	-	2	3	4	.	1	2	1	2	3	4	5	6	7
-	2	3	4	.	1	2									
1	2	3	4	5	6	7									

Видно, что:

- при «усечении» чисел проводится округление;
- знак «+» не выводится (но позиция для него предусмотрена).

Это так называемый «форматированный вывод» с помощью инструкции `print()`.

Если указанное в инструкции значение `КолДробн` больше фактического количества цифр в дробной части выводимого значения, то справа будут выведены дополнительные нули:

Программа	На экран будет выведено						
<pre>a = 1/2 print('% .3f ' % a)</pre>	0.500 <table><tr><td></td><td>0</td><td>.</td><td>5</td><td>0</td><td>0</td></tr></table> Позиция экрана слева 123456		0	.	5	0	0
	0	.	5	0	0		

Несколько слов о параметре `<значение>`. Это должно быть имя переменной:

```
a = 1
b = 3
c = a/b
print("% .1f " % c)
```

или конкретное значение:

```
print('% .2f ' % 0.567743)
```

или арифметическое выражение в скобках:

```
print('% .1f ' % (1/b))
print('% .1f ' % (a/3))
print('% .1f ' % (a/b))
print('% .1f ' % (1/3))
```

Можно установить также общее количество позиций экрана для вывода числа. Это количество `ОбщКол` указывается в формате вывода (между символами `' % '` или `" % "`) до точки:

```
print('% 7.3f ' % <значение>)
```

или

```
print("% 6.2f" % <значение>)
```

Значение **ОбщКол** включает в себя цифры дробной части (**КолДробн**), разделитель-точку и знак числа.

Примеры:

Программа	На экран будет выведено														
<pre>a = 234.193 print('% 6.1f' % a)</pre>	234.2 <table><tr><td></td><td>2</td><td>3</td><td>4</td><td>.</td><td>2</td></tr></table> Позиция экрана слева <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td></tr></table>		2	3	4	.	2	1	2	3	4	5	6		
	2	3	4	.	2										
1	2	3	4	5	6										
<pre>b = -895.451 print('% 7.2f' % b)</pre>	-895.45 <table><tr><td>-</td><td>8</td><td>9</td><td>5</td><td>.</td><td>4</td><td>5</td></tr></table> Позиция экрана слева <table><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr></table>	-	8	9	5	.	4	5	1	2	3	4	5	6	7
-	8	9	5	.	4	5									
1	2	3	4	5	6	7									

Если фактическое количество цифр в выводимом значении меньше, чем **ОбщКол**, оно будет дополнено слева пробелами (а справа – возможно, нулями):

Программа	На экран будет выведено							
b = 30/8 print('% 7.3f'% b)	3.750 <table><tr><td></td><td></td><td>3</td><td>.</td><td>7</td><td>5</td><td>0</td></tr></table> Позиция экрана слева 1 2 3 4 5 6 7			3	.	7	5	0
		3	.	7	5	0		

а если больше, то значение **ОбщКол** игнорируется и в целой части выводится фактическое количество цифр:

Программа	На экран будет выведено																		
<pre>b = 30000/7 print('% 7.3f' % b)</pre>	4285.714 <table><tr><td></td><td>4</td><td>2</td><td>8</td><td>5</td><td>.</td><td>7</td><td>1</td><td>4</td></tr><tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td></tr></table> Позиция экрана слева		4	2	8	5	.	7	1	4	1	2	3	4	5	6	7	8	9
	4	2	8	5	.	7	1	4											
1	2	3	4	5	6	7	8	9											

Формат вывода можно применить к нескольким переменным:

```
print('% 6.2f' % a, '% 6.2f' % b)
```

Можно также обеспечить форматированный вывод целых чисел. В этом случае в формате вывода указывается только общее количество позиций экрана **ОбщКол**, а также буква **d** (говорящая о том, что нужно вывести целое число в десятичной системе счисления):

Программа	На экран будет выведено										
<pre>b = 5 print('% 10d' % b)</pre>	5 <table><tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>5</td></tr></table> Позиция экрана слева 12345678910										5
									5		

Инструкция `print()` с параметром `ОбщКол`, как правило, используется для вывода на экран данных в виде таблиц:

21.55	79.34
-36.40	2.70
128.67	-456.28
3.20	66.38

Возможны и другие виды форматирования выводимых на экран данных (выравнивание на строке и т. п.).

Перед форматированным выводом числа (и/или после него) можно записать также тексты:

```
s = a + b
print('Сумма равна', '% 8d' % s)

dlina = 8123.5/7
print('% 7.1f' % dlina, 'км')
```

Контрольные вопросы

1. Что можно указывать в скобках в инструкции `print()`? Что будет выведено на экран в том или ином случае?
2. Можно ли указывать в скобках несколько значений одного типа? Что при этом будет выведено на экран между ними? Как изменить этот разделитель?
3. Можно ли указывать в скобках несколько значений разного типа?
4. Что надо сделать, чтобы после выполнения инструкции `print()` следующие данные выводились на той же строке?
5. В чем особенность вывода на экран вещественных значений?
6. Как можно ограничить количество цифр в дробной части вещественного числа при его выводе на экран?
7. Что устанавливает значение `ОбщКол`, о котором рассказывалось выше? В каком случае оно, как правило, используется?

Задания

1. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

```
print(31, 15, end = '')
print(77)
```

2. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

```
print(51, 36)
print(77, 45, end = '')
```

3. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих инструкций:

```
print(25, 86)
print()
print(27, 51, end = '')
```

4. Определите (не оформляя программу), что будет выведено на экран в результате выполнения следующих фрагментов программ:

а)
a = 5; b = 3
print('F(', b, ') = ', a, sep = '')

б)
a = 5; b = 3
print('f(a)=', '(b)', sep = ' ')

в)
a = 5; b = 3
print('F(', a, ')=(', b, ')', sep = ' ')

Задачи для разработки программ

1. Получить на экране следующее:

Три числа: 10 5 24

Текст 'Три числа: 10 5 24' не использовать.

2. Получить на экране следующее:

Три числа: 103, 25, 724

Текст 'Три числа: 103, 25, 724' не использовать.

3. Получить на экране следующее:

1

2

3

4. Число π примерно равно 3,1415926. Вывести на экран это число с тремя цифрами в дробной части. Текст '3.142' не использовать.

5. Получить на экране следующее:

```
ooooo124oooooo13
oooooo56oooooo355
ooooo587ooooooo8
```

где символом «o» обозначена пустая позиция на экране.

6. Получить на экране следующее:

```
ooooo1.24oooo13.52
oooo3.567ooo-355.1
oooooo8.2oooooo9.18
```

где символом «o» обозначена пустая позиция на экране.

7. Получить на экране следующее:

```
ooooo7.240ooo-43.520
oooo23.500oooo55.107
```

oooo88.203oo-769.800

где символом «о» обозначена пустая позиция на экране.

В заключение заметим, что в последних версиях Python для форматированного вывода данных используется также метод `format()`. Этот метод форматирует значение-аргумент, указанное в круглых скобках, по шаблону (образцу), который указан в кавычках и фигурных скобках:

```
print('{<шаблон форматирования>}'.format(<значение>))
```

Примеры:

```
print('{: 10.2f}'.format(-4/3))
```

```
a = -4/3
```

```
print('{: 10.2f}'.format(a))
```

```
z = 123
```

```
print('{: 5d}'.format(z))
```

Особенности оформления шаблона вывода:

1) между первыми (левыми) символами '{ ' пробела быть не должно. В противном случае он также будет выведен;

2) после двоеточия пробела может не быть; два и более пробелов записывать нельзя;

3) перед «правыми» символами '}' пробела быть не должно (между ними пробелы возможны);

4) для вещественных чисел в шаблоне указываются значения КолДробн и ОбщКол (или только КолДробн), для целых – ОбщКол.

Метод `format()` также может быть использован и для нескольких значений сразу (список этих значений указывается в скобках через запятую):

```
a = 1/3
```

```
b = 1/9
```

```
print('{: 7.3f}{: 7.3f}'.format(a, b))
```

```
print('{: 5.1f}{: 5.1f}'.format(1/3, 1/7))
```

```
x = 317
```

```
y = 123
```

```
z = 72
```

```
print('{: 5d}{: 5d}{: 5d}'.format(x, y, z))
```

```
print('{: 5d}{: 5d}{: 5d}'.format(13, 5, 41))
```

При этом между отдельными шаблонами вывода {...} пробелов быть не должно. В противном случае они также будут выведены.

Задание

Разработайте программы решения приведенных выше задач 4–7 с использованием метода `format()`

Задачи с простыми числами

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.

Обсудим методику разработки программ решения ряда задач, связанных с простыми числами – натуральными (целыми положительными) числами, не имеющими других делителей, кроме самих себя и единицы. При этом число 1 простым не считается.

Как принято в нашем журнале, мы используем школьный алгоритмический язык (система программирования КуМир). Русский синтаксис этого языка делает приводимые методики и программы максимально понятными, и вы сможете разработать аналогичные программы на языке программирования, который изучаете.

Задача 1. Дано натуральное число, большее 1. Проверить, является ли оно простым

Очевидный метод решения этой задачи очевиден – он определяется в соответствии с определением. Следует определить количество делителей и в зависимости от этого количества вывести ответ.

В приведённой ниже программе использованы следующие переменные величины:

n – заданное число;

$возм_дел$ – его возможные делители;

$кол_дел$ – количество делителей (переменная-счётчик).

Напомним также, что делителями числа n являются числа, на которые n делится без остатка. В школьном алгоритмическом языке для определения остатка от деления одного целого числа на другое предназначена функция *mod* (в других языках программирования для используется не функция, а специальная операция)

Итак, программа:

```
алг Проверка
нач цел  $n$ ,  $возм\_дел$ ,  $кол\_дел$ 
  ввод  $n$ 
   $кол\_дел := 0$  | Начальное значение переменной-счётчика
  нц для  $возм\_дел$  от 2 до  $n - 1$ 
    если  $mod(n, возм\_дел) = 0$ 
      то | Встретился делитель числа  $n$ 
        | учитываем его в количестве делителей
         $кол\_дел := кол\_дел + 1$ 
    все
  кц
  | В зависимости от значения  $кол\_дел$  выводим ответ
  если  $кол\_дел = 0$ 
    то
      вывод нс, "Это число простое"
    иначе
      вывод нс, "Это число составное"
  все
кон
```

Количество проверок (а при большом числе n оно может быть значительным) можно сократить, если учитывать, что простым числом может быть только нечётное число, а у нечётных чисел делителем может быть только нечётное число. Соответствующий фрагмент:

```

нц для возм_дел от 3 до  $n - 1$  шаг 2 | Только нечётные числа
  если mod( $n$ , возм_дел) = 0
    то
      ...

```

Можно также учесть тот факт, что если среди чисел до квадратного корня из n нет делителей, то их нет и среди остальных чисел (убедитесь в этом!). Это значит, что проверять можно до числа, равного целой части квадратного корня из n (почему до целой части?):

```

нц для возм_дел от 2 до int(sqrt( $n$ ))
  если mod( $n$ , возм_дел) = 0
    то
      ...

```

где *sqrt* и *int* – функции школьного алгоритмического языка, возвращающие, соответственно, значение квадратного корня и целой части своего аргумента.

Обратим внимание на то, в качестве начального и конечного значения параметра цикла приняты значения 2 и $n - 1$, а шаг изменения параметра принят равным 1. Это связано с тем, что если оформить

```

нц для возм_дел от 3 до int(sqrt( $n$ )) шаг 2

```

то числа 4, 6 и 8 будут определены как простые (убедитесь в этом и установите – почему?).

Можно также проверку чётных чисел провести отдельно:

```

ввод  $n$ 
если mod( $n$ , 2) = 0
  то
    вывод нс, "Это число составное"
    выход | Выход из программы
все
  кол_дел := 0
  нц для возм_дел от 3 до int(sqrt( $n$ )) шаг 2
    ...

```

Можно также не подсчитывать количество делителей, а рассуждать следующим образом – если встретится хотя бы один делитель, то проверяемое число простым не является. Для фиксации такого факта можно использовать переменную логического типа *прост* (или целого типа, принимающую значения 0 и 1). Соответствующая программа:

```

алг Проверка без подсчёта делителей
нач цел  $n$ , возм_дел, лог прост
  ввод  $n$ 
  прост := да
  нц для возм_дел от 2 до int(sqrt( $n$ ))
    если mod( $n$ , возм_дел) = 0
      то
        вывод нс, "Это число составное"
        прост := нет
        выход | Дальнейший перебор и проверку можно не проводить
  все
кц
если прост = да
  то
    вывод нс, "Это число простое"
все
кон

```

Задача 2. Определить количество простых чисел, не больших 1000000

На основе двух вариантов решения задачи 1 можно разработать два варианта решения указанной задачи (оба – с использованием так называемого «вложенного оператора цикла»):

1) с подсчётом количества делителей:

```
алг Подсчет 1
нач цел n, возм_дел, кол_дел, кол_прост
кол_прост := 0 | Новая переменная – искомое количество
нц для n от 2 до 1000000 | Цикл перебора чисел
кол_дел := 0
нц для возм_дел от 2 до int(sqrt(n)) | Цикл проверки чисел
если mod(n, возм_дел) = 0
то
кол_дел := кол_дел + 1
все
кц
если кол_дел = 0
то | Встретилось очередное простое число
кол_прост := кол_прост + 1
все
кц
вывод нс, кол_прост
кон
```

2) без подсчёта количества делителей:

```
алг Подсчет 2
нач цел n, возм_дел, кол_прост, лог прост
кол_прост := 0
нц для n от 2 до 1000000
прост := да
нц для возм_дел от 2 до int(sqrt(n))
если mod(n, возм_дел) = 0
то
прост := нет
выход
все
кц
если прост = да
то
кол_прост := кол_прост + 1
все
кц
вывод нс, кол_прост
кон
```

Задание для самостоятельной работы

Разработайте программу решения задачи 2 на языке программирования, который вы изучаете. В ней уменьшите (по сравнению с приведёнными программами) количество проверяемых чисел. Ответ и программу пришлите ответ в редакцию. Фамилии всех приславших будут опубликованы, а авторы лучших программ будут награждены дипломами.

В октябре 2024 года был установлен новый рекорд по размеру самого большого установленного простого числа – это число $2^{136\,279\,841} - 1$. В нём более чем 40 миллионов цифр.

Четыре мальчика

Четыре друга, живущие в одном доме, Юнис, Павел, Вазген и Кирилл, учатся в первом, втором, третьем и пятом классах. Историю начинают изучать с пятого класса. Павел учится только на «4» и «5», а все ребята, младше его, стараются брать с него пример. Вазген уже изучает историю. Юнис помогает решать задачи по арифметике мальчику, который младше него. Кто из них в каком классе учится?

Четыре числа

На доске записаны четыре числа: 37 35 33 31. Известно, что в которой каждое следующее число на единицу больше предыдущего. На первый взгляд, это невозможно. Но дело в том, что числа записаны в четырёх разных системах счисления (одно — в восьмеричной). Определите, в какой системе счисления записано последнее число.

У кого какая профессия?

В шахматном турнире принимали участие шесть партнеров разных профессий: токарь, слесарь, инженер, учитель, врач и шофер.

Известно, что:

- 1) в первом туре Андреев играл с врачом, учитель — с Борисовым, а Григорьев — с Евдокимовым; в первом туре 3 партии;
- 2) во втором туре Дмитриев играл с токарем, а врач — с Борисовым; во втором туре 2 партии;
- 3) в третьем туре Евдокимов играл с инженером; в третьем туре 1 партия; тем временем токарь обсуждал с учителем ход игры во втором туре;
- 4) по окончании турнира места распределились так: Борису присудили 1-е место, Григорьев и инженер поделили 2-е и 3-е места, Дмитриев занял 4-е место, а Золотарев и слесарь поделили 5-е и 6-е места.

У кого какая профессия?

Исполнитель Черепаха. Задача 1

Исполнитель Черепаха действует на плоскости с декартовой системой координат. В начальный момент Черепаха находится в начале координат, её голова направлена вдоль положительного направления оси ординат, хвост опущен. При опущенном хвосте Черепаха оставляет на поле след в виде линии. В каждый конкретный момент известно положение исполнителя и направление его движения. У исполнителя существует две команды: **Вперёд n** (где n — целое число), вызывающая передвижение Черепахи на n единиц в том направлении, куда указывает её голова, и **Направо m** (где m — целое число), вызывающая изменение направления движения на m градусов по часовой стрелке.

Запись **Повтори k [Команда1 Команда2 ... КомандаS]** означает, что последовательность из S команд повторится k раз.

Черепахе был дан для исполнения следующий алгоритм: **Повтори 7 [Вперёд 10 Направо 120]**.

Определите, сколько точек с целочисленными координатами будут находиться внутри области, ограниченной линией, заданной данным алгоритмом. Точки на линии учитывать не следует.

Найти двузначное число

Вот задача: «Найти такое двузначное число, которое вдвое больше произведения своих цифр». Её можно решить разработав компьютерную программу или аналитически.

В данной статье опишем методику решения этой задачи средствами электронной таблицы Microsoft Excel или аналогичной программы.

В верхней части листа запишем все двузначные числа (кроме, кратных 10). Для этого оформим лист следующим образом (рис. 1):

	A	B	C	D	E	F	G	H	I	J
1	Числа									
2		1	2	3	4	5	6	7	8	9
3	1	11	12	13	14	15	16	17	18	19
4	2	21	22	23	24	25	26	27	28	29
5	3	31	32	33	34	35	36	37	38	39
6	4	41	42	43	44	45	46	47	48	49
7	5	51	52	53	54	55	56	57	58	59
8	6	61	62	63	64	65	66	67	68	69
9	7	71	72	73	74	75	76	77	78	79
10	8	81	82	83	84	85	86	87	88	89
11	9	91	92	93	94	95	96	97	98	99
12										

Рис. 1

В столбце А указаны все цифры десятков, в строке 2 – все цифры единиц чисел (все цифры целесообразно записать с помощью так называемого «автозаполнения ячеек»). Числа следует записать, естественно, не вручную, а с помощью формул. Для этого следует в ячейку В3 ввести формулу $=A3*10+B\$2$, которая оформлена так, что её можно распространить (скопировать на остальные ячейки диапазона В3:J11 (в ней использованы так называемые «относительные ссылки/адреса» ячеек А3 и В\$2).

Ниже в строках 14-23 получим все попарные произведения цифр записанных чисел (рис. 2). Соответствующие формулы введите самостоятельно.

	A	B	C	D	E	F	G	H	I	J
...										
12										
13	Произведения цифр									
14		1	2	3	4	5	6	7	8	9
15	1	1	2	3	4	5	6	7	8	9
16	2	2	4	6	8	10	12	14	16	18
17	3	3	6	9	12	15	18	21	24	27
18	4	4	8	12	16	20	24	28	32	36
19	5	5	10	15	20	25	30	35	40	45
20	6	6	12	18	24	30	36	42	48	54
21	7	7	14	21	28	35	42	49	56	63
22	8	8	16	24	32	40	48	56	64	72
23	9	9	18	27	36	45	54	63	72	81
24										

Рис. 2

Осталось проверить, равны ли числа на рис. 1 удвоенному произведению их цифр на рис. 2. Такую проверку проведём в диапазоне B27:J35. Для этого удобно использовать функцию ЕСЛИ (см. ниже дополнение к статье). Её оформите так, чтобы в ячейке с искомым числом появилось это число, а остальные ячейки указанного диапазона были пустыми (рис. 3).

	A	B	C	D	E	F	G	H	I	J
...										
24										
25	Сравнение									
26		1	2	3	4	5	6	7	8	9
27	1									
28	2									
29	3									
30	4									
31	5		52							
32	6									
33	7									
34	8									
35	9									
36										

Рис. 3

Примечание. Значение 52 на рис. 3 указано условно.

Задачи для самостоятельного решения

1. Определите, сколько существует двузначных чисел, которые в три раза больше произведения своих цифр. Какие это числа?
2. Определите, сколько существует двузначных чисел, которые на 5 больше или на 5 меньше утроенной суммы их цифр. Какие это числа?

Листы с решением всех трёх присылайте в редакцию. Фамилии всех приславших будут опубликованы, а авторы лучших работ будут награждены дипломами.

Дополнение

О функции ЕСЛИ

Функция ЕСЛИ используется, когда в какой-то ячейке возможны два варианта значений, а выбор того или иного значения зависит от некоторого условия.

Общий вид функции:

ЕСЛИ(<условие>; <значение_1>; <значение_2>)

В простейшем случае аргумент <условие> — это сравнение конкретных значений (чисел, текста и др.), адресов ячеек и выражений с использованием знаков "=", ">", ">=", "<", "<=" и "<>":

A10 = 100
 B12 < B11
 B12 + B13 <> 0
 C44 = "яблоки"
 D10 + D11 <= E10 + E11

Если <условие> — истинное, то в ячейке с функцией ЕСЛИ будет отображен второй аргумент (значение_2), если ложное — третий аргумент (значение_2). Например, функция:

ЕСЛИ(A1>10, "Больше 10", "10 или меньше")

возвращает строку "Больше 10", если значение в ячейке A1 больше 10, и "10 или меньше", если оно меньше или равно 10.

Если в формуле не указать третий аргумент, то в случаях ложности заданного условия в ячейке будет выведено значение ЛОЖЬ. Например, такой результат будет выведен при формуле:

=ЕСЛИ(D12 > 0; D12 * D12)

и значении в ячейке D12, равном –2.

Если же формулу оформить так:

=ЕСЛИ(D12 > 0; D12 * D12;)

то в указанном случае будет выведен 0.

В условии в функции ЕСЛИ могут быть использованы не только простые условия (см. выше), но и так называемых «сложные условия». Они формируются с помощью функций И и ИЛИ. В качестве аргументов этих функций можно указывать два или больше простых условия:

И(D12 > 0; E12 < 0)

ИЛИ(A10 = 5; B10 = 5; C10 = 5)

Результат, возвращаемый этими функциями, определяется по известным вам таблицам истинности для логических операций конъюнкции и дизъюнкции.

Книжная полка

Полезная книга

Издательство ДМК Пресс предлагает книгу «1400 задач по программированию»
<https://dmkpress.com/catalog/computer/handbooks/978-5-97060-827-2/>



Если вы, уважаемый 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. В приобретении такого опыта вам поможет данная книга.

В книге приведено более 1400 задач по программированию. Задачи имеют разный уровень сложности и охватывают большой круг тем. Они не привязаны к конкретному языку программирования и могут быть решены в программе на любом языке (Pascal, Python, C, школьном алгоритмическом языке и т. д.).

Через сайт издательства книгу можно приобрести дешевле, чем в магазинах (даже с учетом стоимости доставки).

Еще раз об использовании двоичной системы счисления в докомпьютерную эпоху

Д.М. Златопольский, Москва

В предыдущем выпуске нашего журнала была опубликована наша статья, в которой приведён ряд примеров использования двоичной системы счисления до момента создания электронных вычислительных машин. Уже после её публикации мы нашли информацию о новых результатах исследования архива великого немецкого полимата⁴ Готфрида Вильгельма Лейбница (1646–1716). В частности, в Государственной библиотеке Нижней Саксонии (Niedersächsische Landesbibliothek), г. Ганновер, ФРГ, была найдена его рукопись, относимая примерно к 1716 году (см. рис. 1).



Рис. 1

Расшифровка рукописи сотрудниками библиотеки Siegmund Probst и Julia Weckend показала, что текст и рисунки в ней относятся к часам оригинальной конструкции. В них имеется всего одна стрелка. Между каждым из 12 чисел (1, 2, ..., 12) и центром циферблата находится ряд из четырёх «точек», каждая из которых либо полая, либо выпуклая. Любая полая точка соответствует нулю, а выпуклые:

- первая (самая близкая к центру) – единице;
- вторая – двум;
- третья – четырём;
- четвёртая (на периферии циферблата) – восьми.

На рис. 2 показано схематическое изображение часов, подготовленное Lisa Silva.

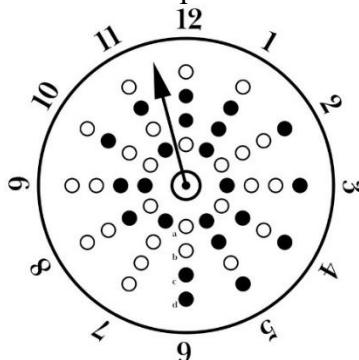


Рис. 2

⁴ Полимат (*polymathēs*, греч. πολυμαθής – от *πολυ* – «много» и *μαθής* – «занятия») – «универсальный человек», энциклопедист, человек эпохи Возрождения, тот, чьи интеллектуальные способности, интересы и деятельность не ограничены одной областью знаний и единственной областью их применения.

Полые точки обозначены «пустыми» кружками, выпуклые – закрашенными. Перечисленные чуть выше четыре точки обозначены буквами a , b , c и d .

Идея состоит в том, что если нащупать, куда указывает стрелка, затем провести пальцем от кончика часовой стрелки вниз к центру, а затем дальше вниз по той же линии, пока не встретится ряд из четырёх точек, которые затем считываются наощупь. Например, ряд в нижней части циферблата имеет выпуклые точки c и d , сумма «значений» которых равна $4 + 8 = 12$ часов.

Иными словами, Лейбниц нарисовал схему тактильных часов, которые позволили бы определять время ночью или слепым людям. Причём можно сказать, что на часах Лейбница информация представлена в виде четырёхразрядных двоичных чисел ($12 = 1100_2$, $1 = 0001_2$, $5 = 0101_2$ и т.п.).

Внимательный читатель, конечно, обнаружил на циферблате несоответствие между числами и их двоичными представлениями. Причина этого в том, что циферблат предназначен не для того, чтобы его можно было увидеть, а только потрогать: ряд точек, который можно «прочитать», находится прямо напротив того места, куда указывает часовая стрелка, а не тот, который был бы под стрелкой (к которому в любом случае нельзя будет прикоснуться, потому что над ним будет часовая стрелка).

Тактильные часы Лейбница столь же просты, сколь и гениальны, и эта идея тем более примечательна, что она более чем на столетие предшествует родственным изобретениям – так называемым «ночному письму»⁵ и «шрифту Брайля»⁶.

В рукописи Лейбниц ничего не говорит о механизме часов, но часовщику того времени было бы достаточно легко сконструировать его. Тактильная конструкция часов, безусловно, жизнеспособна, хотя, возможно, уязвима для неуклюжего или сильного прикосновения к часовой стрелке.

Были ли изготовлены предложенные Лейбницем часы – неизвестно (переписка учёного с инженером Готфридом Тойбером позволяет предположить, что его намерением было, чтобы Тойбер изготовил такие часы).

Автор выражает благодарность Лойду Стрикленду (Lloyd Strickland), профессору Департамента истории, политики и философии Манчестерского университета, Великобритания, за предоставленную информацию, использованную при подготовке данной статьи.

Литература

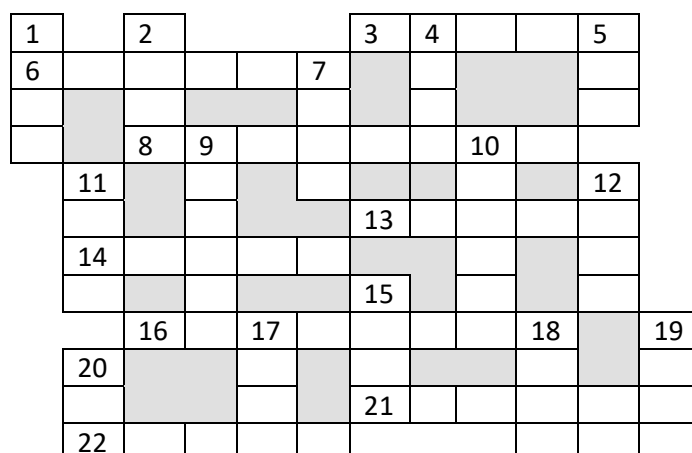
1. *Lloyd Strickland. Leibniz on Number Systems. In Bharath Sriraman (ed.), Handbook of the History and Philosophy of Mathematical Practice. Cham: Springer. pp. 1-31 (2023)*

⁵ «Ночное письмо» (фр. *écriture nocturne*) – система кодирования, разработанная в 1819 году Шарлем Барбье. В ней применялись выпуклые точки и тире на плотной бумаге, и предлагалась она солдатам как средство беззвучной ночной связи в полевых условиях. Солдаты должны были продавливать точки и тире с обратной стороны бумаги пером, похожим на шило. Получатель такого сообщения читал эти выпуклости, водя по ним кончиками пальцев.

⁶ Шрифт Брайля – рельефно-точечная система письма для слепых (см. предыдущий выпуск журнала «Мир информатики»).

Кроссворд

Решите, пожалуйста, кроссворд.



По горизонтали

3. Один из контактов транзистора типа **МОП** (**М**еталл — **О**ксид — **П**олупроводник).
6. Рассмотрение и при необходимости обработка всех элементов массива.
8. Изображение чего-нибудь, рассказ о ком-нибудь (о чем-нибудь) в письменной или устной форме.
13. Поименованная группа файлов, объединенных по какому-то признаку.
14. Процесс написания текста на компьютере.
16. Совокупность четко определенных правил для решения задачи за определенное число шагов.
21. Указатель места на экране монитора.
22. Так иногда называют столбец в таблице.

По вертикали

1. Непрошенное рекламное сообщение, сетевой “мусор”.
2. Язык программирования (как правило, для начинающих изучать последнее).
4. Жаргонное название результата обработки изображения на устройстве ввода информации в компьютер.
5. Число в системе условных обозначений символов.
7. Разновидность носителя информации.
9. Название клавиши.
10. Ввоз товаров из-за рубежа, а также вставка в документ приложений Windows объектов из других приложений.
11. Часть экрана, занимаемая приложением или документом Windows.
12. ... данных.
15. Синоним слова “дорожка” (участка магнитного диска).
17. Конечное число точек на плоскости, соединенных отрезками кривых линий.
18. Устройство для соединения двух участков локальной сети.
19. Знак препинания.
20. Элемент языка разметки гипертекста.

Изменить имя

Посмотрите на составленное из спичек мужское имя — Глеб. Нужно превратить его в другое мужское имя, переставив всего две спички. Как?



Математические кроссворды

О классических кроссвордах со словами, в которых неизвестные слова там вписывают по горизонталям слева направо и вертикалям сверху вниз, знают все. На сайтах с головоломками можно встретить также числовые кроссворды (их также называют «кросснамберами» - от английского слова *number* – число). В них предлагается заполнить сетку числами, приведёнными вперемешку. А вот математические кроссворды встречаются намного реже. Предлагаем решить два таких кроссворда:

1.

1	×	2	
–			
3		+	4
			+
		5	–

По горизонтали:
1. 42
3. 57
5. 3

По вертикали:
1. 5
2. 172
4. 3

2.

1	÷	2	+	3
			×	
4		+		
×				
5	×		+	

По горизонтали:
1. 12
4. 127
5. 32

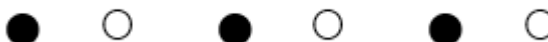
По вертикали:
2. 43
3. 56
4. 18

Правила такие: «Нужно заполнить белые клетки цифрами из диапазона от 1 до 9. В каждом арифметическом выражении в сетке все цифры должны быть различны. Результаты выражений по горизонтали и вертикали приведены под кроссвордами».

Использованы материалы журнала «Квантик», автор – Риад Ханмагомедов

Шесть шашек

На столе лежат 6 шашек в ряд попеременно — чёрная, белая, чёрная и т.д.:



Справа и слева имеется свободное место, достаточное для размещения четырёх шашек. Требуется за минимальное число перемещений переместить шашки так, чтобы на этих шести местах слева оказались все белые, а вслед за ними все чёрные шашки. При этом перемещать на свободное место нужно сразу две шашки, не меняя порядка, в котором они лежат.

Алгоритм решения задачи оформите в виде:

Исходное состояние	____ Ч Б Ч Б Ч Б ____
Ход 1	__ Ч Б Ч Б __ Ч Б ____

...	
-----	--

Шесть чашек

Представьте ряд из шести чашек. Первые три – пустые, а три следующие наполнены водой. Как за минимально возможное число касаний чашек добиться чередования пустых чашек и чаше с водой? Толкать чашку чашкой запрещено.

Чему равен треугольник?⁷

$$\blacksquare + \blacksquare = 20$$

$$\blacksquare \times \heartsuit + \heartsuit = 22$$

$$\blacksquare \times \heartsuit - \blacktriangle \times \blacksquare = \blacksquare$$

$$\blacktriangle = ?$$

Не опечатка ли?

На одном математическом конкурсе в 2001 году предлагалась такая задача: «В книге рекордов Гиннеса написано, что наибольшее известное простое число равно $23021^{377} - 1$. Не опечатка ли это?».

Ваше мнение? Почему?

Восемь вопросов и один термин

Ответив на восемь вопросов и вписав ответы в столбцы, в выделенном горизонтальном ряду прочтите слово — одно из основных понятий информатики.

1	2	3	4	5	6	7	8

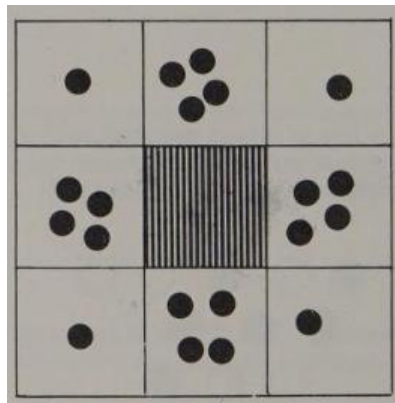
1. Основной структурный элемент презентации, создаваемой с помощью программы Microsoft PowerPoint.
 2. Устройство с органами управления, с помощью которых оператор воздействует на управляемые объекты.
 3. Размер (высота) шрифта.
 4. Последовательность букв и цифр, ограниченная с обоих концов пробелами, запятыми, точками, дефисами и т.п.
 5. Программа, обладающая способностью к самовоспроизведению.
 6. Совокупность характеристик символа или абзаца.
 7. Непрерывная последовательность данных.
 8. Результат арифметической операции.
- Ответы (9 слов, найденное — с комментариями к нему) присылайте в редакцию.

⁷ Задача предназначена для учеников 1–7-х классов.

Две задачи о заключенных

Задача 1. 20 заключенных

Однажды в древности начальник одной из тюрем, будучи любителем математики, приказал надзирателям разместить 20 заключённых в восьми камерах так, чтобы в каждом из четырёх рядов из трёх камер было одно и то же число заключённых – 6. Надзиратели выполнили задание так, как показано на рисунке:



Через некоторое время он приказал разместить тех же 20 заключённых в восьми камерах так, чтобы в каждом из четырёх рядов из трёх камер было одно и то же число заключённых – 7. Возможно ли это?

Задача 2. Две двери

Однажды в древности отважного воина захватили слуги султана. Султан приказал отправить смельчака в темницу, где вместе с ним непрерывно будут рядом два стражника. Один стражник исключительно правдив в разговоре, а второй всегда лжёт. Внутри темницы оказались два закрытых входа. Войдёшь в один – выберешь «неволю навсегда», а в другой – «освобождение». Султан предложил воину избрать один вход: если это окажется «освобождение», то пленник свободен.

Воину позволили спросить надзирателей только один раз. То есть один вопрос на двоих надзирателей. Пленник не в курсе, кто из них лжец, а кто правдив. Надзиратели, несомненно, знают, какой вход подарит воину освобождение.

Какой вопрос воин должен задать одному из стражников для достоверного определения входа, который означает «освобождение»?

Заполнить квадрат

Расставьте в каждую клетку квадрата:

- числа 1, 2, 3, 4;
- буквы А, В, С, D;
- цвета: жёлтый, зелёный, синий и коричневый (в виде строчных букв ж, з, с, к)

так, чтобы в каждой строке и в каждом столбце все клетки были разного цвета и чтобы не повторялись пары «число – буква», «буква – цвет» и «число – цвет».

Некоторые элементы уже стоят на своих местах.

			4Dж
		4Ac	
	4Cк		
4Bз			

Конкурс № 74 «Пять простых чисел»

В качестве задания этого конкурса редакция предлагает решить задачу: «Есть пять простых чисел. Сумма трёх из них равна 100, а четырёх – 1000. Какова сумма всех пяти чисел? Сколько существует наборов из пяти простых чисел, удовлетворяющих указанным особенностям? Сколько различных чисел в них представлено? Какие это числа?»

Ответы присылайте по адресу: mirinfo@infojournal.ru. Укажите в письме свои фамилию и имя, населенный пункт, номер школы и фамилию, имя и отчество учителя информатики. Срок представления ответов — 31 октября 2025 г.

Можно отвечать не на все вопросы.

Комментарии к выполнению

Найдите в Интернете список простых чисел, не превышающих сначала 100, затем – 1000.

Рассмотрите все наборы из двух простых чисел a и b и найдите число, которое вместе с ними даёт в сумме 100. Исследуйте, является ли это третье число с простым. Для анализа удобно использовать электронную таблицу Microsoft Excel или подобную программу:

	A	B	C
1	a	b	c
2	1	2	=100-A2-B2
3	3	2	
4	5	2	

Два вопроса

1. Этот литературный герой, вернувшись в декабре 1716 года домой, стал проводить каждый день с утра до вечера в обществе лошадей на своей конюшне, которых самолично кормил и о которых заботился. Связано это было с полным разочарованием в людях. Кто этот герой?

2. Этот классик советской литературы в старости страдал от хронического безденежья. Но, будучи натурой ироничной, он говорил друзьям, что ему впору писать новый вариант повести, за которую Эрнеста Хемингуэя удостоили Нобелевской премии. И озаглавить её в связи с плачевными обстоятельствами жизни так: “Старик и море... долгов”. Кто этот писатель?

Ответы присылайте в редакцию.

.....

Уважаемые читатели!

Сообщаем, что очередные номера нашего интернет-журнала публикуются в период с 15 по 20 число каждого месяца, кроме июля.