

«Мир информатики»

Журнал для тех, у кого информатика – любимый школьный предмет

Выпуск № 108, февраль 2026 г.

Семинар

Перевод чисел в системах счисления с основаниями, равными степени двойки

Для перевода числа из одной недесятичной системы счисления в другую в общем случае понадобится промежуточный перевод исходного числа в десятичную систему. Но в случаях, когда основания используемых систем равны 2, 4, 8 и 16, такой перевод можно не проводить.

1. Перевод чисел из четверичной системы счисления в двоичную и обратно

1.1. Для перевода целого четверичного числа в двоичную систему счисления необходимо:

1) каждую цифру заданного числа, рассматривая ее как десятичную, представить в виде двухразрядного двоичного числа (может быть — с начальным нулем или состоящим из двух нулей);

2) из полученных двухразрядных двоичных чисел сформировать единое число (без возможного начального нуля), которое и будет являться искомым результатом.

Пример. Дано четверичное число 1023. Перевести его в двоичную систему счисления.

Решение

1)

Цифры заданного числа	1	0	2	3
Соответствующее двоичное число	01	00	10	11

2) результат равен 1001011_2 .

1.2. Для перевода целого двоичного числа в четверичную систему счисления необходимо:

1) начиная с правых разрядов, разбить цифры заданного числа на группы из двух цифр (в крайней слева группе может оказаться одна цифра);

2) число, образованное цифрами в каждой группе, перевести в десятичную систему (так как при этом не могут получиться числа, большие 3, то их можно рассматривать как цифры четверичной системы счисления);

3) из полученных на предыдущем этапе чисел сформировать единое число, которое и будет являться искомым результатом.

Пример. Дано двоичное число 111001001. Перевести его в четверичную систему счисления.

Решение

1)

Группы цифр заданного числа	1	11	00	10	01
Соответствующее четверичное число	1	3	0	2	1

2) результат равен 13021_4 .

2. Перевод чисел из восьмеричной системы счисления в двоичную и обратно

2.1. Для перевода целого восьмеричного числа в двоичную систему счисления необходимо:

1) каждую цифру заданного числа, рассматривая ее как десятичную, представить в виде трехразрядного двоичного числа (может быть — с начальным нулем, с начальными нулями или состоящим из трех нулей);

2) из полученных двоичных чисел сформировать единое число (без возможных начальных нулей), которое и будет являться искомым результатом.

Пример. Дано восьмеричное число 3061. Перевести его в двоичную систему счисления.

Решение

1)

Цифры заданного числа	3	0	6	1
Соответствующее двоичное число	011	000	110	001

2) результат равен 11000110001_2 .

2.2. Для перевода целого двоичного числа в восьмеричную систему счисления необходимо:

1) начиная с правых разрядов, разбить цифры заданного числа на триады — группы из трех цифр (в крайней слева группе может оказаться одна или две цифры);

2) число, образованное цифрами в каждой группе, перевести в десятичную систему (так как при этом не могут получиться числа, большие 7, то их можно рассматривать как цифры восьмеричной системы счисления);

3) из полученных на предыдущем этапе чисел сформировать единое число, которое и будет являться искомым результатом.

Пример. Дано двоичное число 11111001101. Перевести его в восьмеричную систему счисления.

Решение

1)

Группы цифр заданного числа	11	111	001	101
Соответствующее восьмеричное число	3	7	1	5

2) результат равен 3715_8 .

3. Перевод чисел из шестнадцатеричной системы счисления в двоичную и обратно

3.1. Для перевода целого шестнадцатеричного числа в двоичную систему счисления необходимо:

1) каждую цифру заданного числа, рассматривая ее как десятичную, представить в виде четырехразрядного двоичного числа (может быть — с начальным нулем, с начальными нулями или состоящим из четырех нулей);

2) из полученных двоичных чисел сформировать единое число (без возможных начальных нулей), которое и будет являться искомым результатом.

Пример. Дано шестнадцатеричное число 3A1C. Перевести его в двоичную систему счисления.

Решение

1)

Цифры заданного числа	3	A	1	C
Соответствующее двоичное число	0011	1010	0001	1100

2) результат равен 11101000011100_2 .

3.2. Для перевода целого двоичного числа в шестнадцатеричную систему счисления необходимо:

1) начиная с правых разрядов, разбить цифры заданного числа на тетрады — группы из четырех цифр (в крайней слева группе может оказаться одна, две или три цифры);

2) число, образованное цифрами в каждой группе, перевести в десятичную систему. Так как при этом не могут получиться числа, большие 15, то их можно рассматривать как цифры шестнадцатеричной системы счисления (при наличии чисел, больших 9, их следует записать с использованием обозначений для соответствующих цифр — А, В, ..., F);

3) из полученных на предыдущем этапе цифр сформировать единое число, которое и будет являться искомым результатом.

Пример. Дано двоичное число 1010111001101. Перевести его в шестнадцатеричную систему счисления.

Решение

1)

Группы цифр заданного числа	1	0101	1100	1101
Соответствующее шестнадцатеричное число	1	5	C (12)	D (13)

2) результат равен $15CD_{16}$.

4. Перевод чисел из шестнадцатеричной системы счисления в четверичную и обратно

4.1. Для перевода целого шестнадцатеричного числа в четверичную систему счисления необходимо:

1) каждую цифру заданного числа представить в виде двухразрядного четверичного числа (может быть — с начальным нулем или состоящим из двух нулей);

2) из полученных двухразрядных четверичных чисел сформировать единое число (без возможного начального нуля), которое и будет являться искомым результатом.

Пример. Дано шестнадцатеричное число 5E0C. Перевести его в четверичную систему счисления.

Решение

1)

Цифры заданного числа	5	E (14)	0	C (12)
Соответствующее четверичное число	11	32	00	30

2) результат равен 113320030_4 .

4.2. Для перевода целого четверичного числа в шестнадцатеричную систему счисления необходимо:

1) начиная с правых разрядов, разбить цифры заданного числа на группы из двух цифр (в крайней слева группе может оказаться одна цифра);

2) четверичное число, образованное цифрами в каждой группе, перевести в шестнадцатеричную систему (при наличии чисел, больших 9, их следует записать с использованием обозначений для соответствующих цифр — А, В, ..., F);

3) из полученных на предыдущем этапе чисел сформировать единое число, которое и будет являться искомым результатом.

Пример. Дано четверичное число 3021023. Перевести его в шестнадцатеричную систему счисления.

Решение

1)

Группы цифр заданного числа	3	02	10	23
Соответствующее шестнадцатеричное число	3	2	4	B (11)

2) результат равен $324B_{16}$.

Обращаем внимание на то, что при переводе чисел из четверичной системы в восьмеричную и обратно и из восьмеричной системы в 16-ичную и обратно описанную методику применять нельзя.

Описанную методику перевода можно применить к системам счисления с основаниями p и q в случаях, когда p является степенью q или наоборот.

Задания для самостоятельной работы

1. Определите, какое из чисел больше — двоичное 11010110 или восьмеричное 412, не переводя каждое из чисел полностью в другую систему.

2. Не используя перевод в десятичную систему, переведите шестнадцатеричное число ABCD в восьмеричную систему счисления.

3. В 25-ичной системе счисления для обозначения чисел от 0 до 24 будем использовать соответственно цифры 0, 1, 2, ..., 8, 9, A, B, C, D, E, F, G, H, I, J, K, L, M, N и P (последняя применяется вместо буквы O, похожей на 0). Не используя перевод в десятичную систему, переведите:

1) 25-ичное число 3L0A в пятеричную систему счисления;

2) пятеричное число 32061243 в 25-ичную систему счисления.

4. Даны два числа, записанные в 100-ичной системе счисления. Предложите методику расчета их суммы, в которой таблица сложения в указанной системе не используется.

5. Установите, существует ли треугольник, длины сторон которого выражаются числами 12_8 , 11_{16} и 1101_2 ?

Личности

Фредерик Брукс

Фредерик Брукс (1931-2022) — американский учёный в области теории вычислительных систем. Руководил разработкой группой операционных систем для семейства компьютеров OS/360 в фирме IBM, начиная с 1964 года.

В 1999 году награждён самой престижной премией в информатике — Премией Тьюринга — “за исторически значимый вклад в архитектуру компьютеров, операционные системы и инженерии программного обеспечения”.

В 1975 году Ф.Брукс написал книгу “Мифический человек-месяц, или Как создаются программные системы” об управлении проектами в области разработки программного обеспечения. Отрывками из этой книги мы хотим ознакомить наших читателей.



«Эпилог. Пятьдесят лет удивления, восхищения и радости

В моей памяти все еще живы удивление и восторг, с которым я — мне тогда было 13 лет — читал отчет от 7 августа 1944 года об освящении компьютера Mark I, архитектором которого был Говард Айкен (Howard Aiken), а проектировщиками — инженеры Клер Лейк (Clair D. Lake), Бенджамин Дурфи (B. M. Durfee) и Фрэнсис Гамильтон (F. E. Hamilton). Такой же вызывающей

ощущение чуда была статья Ванневары Буша (Vannevar Bush) “That We May Think” в апрельском 1945 года номере “Atlantic Monthly”, в которой он предложил организовать знания в виде огромной гипертекстовой паутины и обеспечить пользователей машинами для переходов по существующим ссылкам и создания новых ассоциативных следов.

Новый толчок моя страсть к компьютерам получила в 1952 году, когда, работая летом на IBM в Эдинкоте, штат Нью-Йорк, я получил практический опыт программирования для IBM 604 и формальное обучение программированию для IBM 701, их первой машины с хранимой программой. Аспирантура у Айкена и Иверсона в Гарварде сделала реальностью мои мечты о профессии, и я связал с ней всю свою жизнь. Немногим Бог дает право зарабатывать на жизнь тем, чем они с радостью занимались бы по собственной воле, по увлечению. Я благодарен судьбе.

Для человека, влюбленного в компьютеры, трудно было бы придумать иное время, когда так радостно было жить. От механических устройств до вакуумных ламп, транзисторов и интегральных схем шло бурное развитие технологии. Первый компьютер, на котором я работал сразу после выпуска из Гарварда, был суперкомпьютер IBM Stretch. Этот компьютер царствовал над миром как самый быстрый с 1961 по 1964 годы; было изготовлено 9 экземпляров. Мой сегодняшний Macintosh Powerbook не только быстрее, с большей памятью и большим диском, но и в тысячу раз дешевле (в пять тысяч раз дешевле с учетом инфляции). Мы были свидетелями того, как поочередно произошли компьютерная революция, революция электронных компьютеров, революция миникомпьютеров и революция микрокомпьютеров, в результате каждой из которых компьютеров становилось на порядки больше.

Радости профессии

Почему заниматься программированием интересно? Какими радостями вознаграждаются те, кто им занимается?

Во-первых, это просто радость, получаемая при создании чего-либо своими руками. Как ребенок радуется, делая куличики из песка, так и взрослый получает удовольствие, создавая какие-либо вещи, особенно если сам их и придумал. Я думаю, что этот восторг — отражение восторга Господа, творящего мир, восторга, проявляющегося в индивидуальности и новизне каждого листочка и каждой снежинки.

Во-вторых, это удовольствие создавать вещи, которые могут быть полезны другим людям. Глубоко в душе мы испытываем потребность в том, чтобы другие использовали результаты нашего труда и считали их полезными. В этом отношении программная система по своей сути — то же, что и изготовленная ребенком подставка для карандашей “папе в подарок”.

В-третьих, это очарование создания сложных головоломных объектов, состоящих из взаимодействующих движущихся частей и наблюдения за их работой, круг за кругом демонстрирующей результаты изначально заложенных принципов. Компьютер с работающей на нем программой обладает доведенным до высшего предела очарованием игорного или музыкального автомата.

В-четвертых, это радость, получаемая от неизменного узнавания нового, проистекающего из неповторимой природы задачи. В том или ином отношении задача всегда ставится по-новому, и тот, кто ее решает, получает новые знания — либо практические, либо теоретические, либо те и другие вместе.

Наконец, наслаждение доставляет работа со столь податливым материалом. Программист, подобно поэту, работает почти непосредственно с чистой мыслью. Он строит свои замки в воздухе и из воздуха, творя силой воображения. Трудно найти другой материал, используемый в творчестве, который столь же гибок, прост для шлифовки или переработки и доступен для воплощения грандиозных замыслов. Однако программная конструкция, в отличие от поэтических творений, реальна, в том смысле, что она движется и работает, производя видимые результаты, которые отделимы от самой конструкции. Она печатает результаты, рисует картинки, производит звуки, приводит в движение рычаги. В наше время осуществилось волшебство мифа и легенды. С клавиатуры вводится верное заклинание, и экран монитора оживает, показывая то, чего никогда не было и не могло быть.

Таким образом, программирование доставляет удовольствие, поскольку отвечает глубокой внутренней потребности в творчестве и удовлетворяет чувственные потребности, которые есть у всех нас.

Печали профессии

Не все, однако, в радость, и если предвидеть присущие этому ремеслу огорчения, то они легче переносятся.

Во-первых, необходима безошибочная точность действий. В этом отношении компьютер также напоминает волшебство. Один неверный знак, одна пауза в заклинании, и чудо не состоялось. Человеку несвойственно совершенство, и оно является необходимым лишь в немногих сферах его деятельности. Мне кажется, что при освоении программирования труднее всего привыкнуть к требованию совершенства.

Кроме того, постановка задач, обеспечение ресурсами и предоставление информации осуществляется другими людьми. Редко удается контролировать условия работы и даже ее цели. На языке администрирования это означает, что полномочия ниже ответственности. Впрочем, похоже, что в любой работе, где должен быть получен результат, формальная власть никогда не соизмерима с ответственностью. На практике фактическая (в противоположность формальной) власть приобретается в результате успешного выполнения задач.

Зависимость от других имеет особенно неприятную системную сторону. Он находится в зависимости от программ, написанных другими людьми, и эти программы зачастую плохо спроектированы, слабо написаны, получены в неполном виде (без исходного текста и контрольных примеров) и плохо документированы. Поэтому программисту приходится тратить многие часы на изучение и исправление вещей, которые, в идеале, должны быть полными, доступными и годными к использованию.

Следующий “минус” связан с тем, разработка грандиозных идей — это удовольствие, а поиск паршивых маленьких “жучков”¹ — это всего лишь работа. В каждом творческом деле бывают ужасные периоды однообразного и кропотливого труда, и программирование не является исключением.

Далее оказывается, что при отладке программы сходимость является линейной, если не хуже, хотя можно было предполагать некое квадратичное приближение к окончанию. В итоге отладка продолжается долго, причем на поиск последних более сложных ошибок уходит больше времени, чем на отыскание первых.

Последняя горесть, а часто и последняя капля, — то, что продукт, на который было положено столько труда, оказывается устаревшим в момент его завершения (или даже раньше). Коллеги и конкуренты уже с пылом работают над новыми и лучшими идеями. И уничтожение плода вашей мысли уже не только задумано, но и запланировано.

На самом деле положение обычно лучше, чем кажется. В то время как ваш продукт уже завершен, этот новый и лучший продукт, как правило, отсутствует на рынке, о нем лишь много разговоров, и для его разработки потребуются месяцы. Настоящий тигр не пара бумажному, если требуется реальное использование. Реальное существование имеет преимущества.

Конечно, технологическая основа разработки всегда развивается. Как только разработка проекта закончена, он становится устаревшим в смысле заложенных в нем концепций. Но для осуществления реального проекта необходимо разбиение на стадии и уровни. Судить о том, является ли некая реализация устаревшей, можно лишь сравнивая ее с другими существующими реализациями, а не с нереализованными идеями. Трудность и цель состоят в том, чтобы найти реальные решения для реальных задач в установленные сроки, используя имеющиеся ресурсы.

Таково программирование — и смоляная яма, в которой увязли многие проекты, и творчество со своими радостями и печалью. Для многих радости значат гораздо больше, чем печали. Для них и написана эта книга в попытке проложить какие-то мостки через это болото».

¹ Автор имеет в виду процесс поиска и устранения ошибок в программе — ее отладку (по-английски *debugging* — от слова *bug* — жук).. — Прим. ред.

Строим циклоиду

Помните оранжевые пластмассовые катафоты — светоотражатели, прикрепляющиеся к спицам велосипедного колеса? Прикрепим катафот к самому ободу колеса и проследим за его траекторией при движении велосипеда. Кривая линия, по которой будет перемещаться катафот, в математике называется “циклоида” (от греческого слова *κυκλοειδής* — *круглый*). Итак, циклоида — это траектория некоторой фиксированной точки окружности, катящейся без скольжения по прямой (см. *рис. 1*). Окружность (в случае велосипеда — колесо) называется “производящей окружностью”.

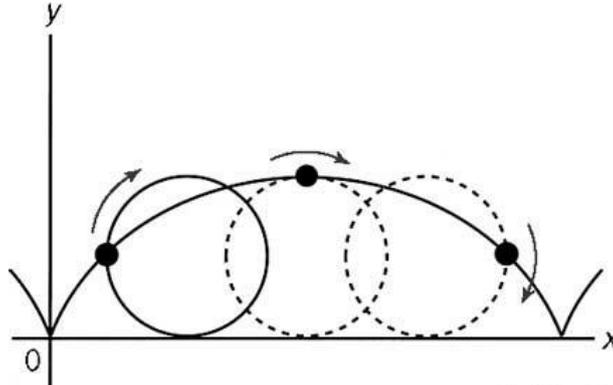


Рис. 1

У циклоиды масса любопытнейших свойств. Оказывается, например, что циклоида является кривой наибыстрейшего спуска. Иначе говоря, скатываясь по снежной горке, профиль которой выполнен в виде перевернутой циклоиды, мы окажемся у основания горки быстрее, чем в случае другой формы горки. Кроме того, циклоида является такой кривой, по которой должна двигаться тяжелая материальная точка, чтобы период ее колебаний не зависел от амплитуды колебаний. Используя это свойство, в XVII веке Христиан Гюйгенс (голландский механик, физик, математик, астроном и изобретатель) сконструировал часы, изображенные на *рис. 2*. Любопытно, что траектория конца маятника, как и ограничивающие его боковые “щеки”, представляет из себя циклоиду.



Рис. 2

Обращали ли вы внимание на то, куда полетит камень, застрявший в протекторе велосипеда или мотоцикла, когда выскочит из него — против направления движения или по направлению? Как известно, свободное движение тела начинается по касательной к той траектории, по которой оно двигалось. Касательная к циклоиде всегда направлена по направлению движения и проходит через верхнюю точку производящей окружности. По направлению движения полетит и наш камушек. Наверняка многие знают, что будет, если кататься по лужам на велосипеде без заднего крыла. Мокрая полоска на спине этих читателей и была житейским подтверждением только что полученного результата.

Теперь перейдем к информатике (☺) и опишем методику получения изображения циклоиды на экране монитора компьютера.

Для решения задачи используем так называемые “параметрические уравнения” линии. Для циклоиды они такие:

$$x = rt - r\sin(t);$$

$$y = r - r\cos(t).$$

где t — угол поворота.

Напомним, что параметрическими такие уравнения называются потому, что они определяют значения координат x и y каждой точки кривой в зависимости от некоторого параметра, в нашем случае от параметра t . Кроме того, в уравнениях есть величина r — радиус производящей окружности.

Из *рис. 1* видно, что циклоида состоит из одинаковых участков, то есть, как говорят в математике, “является периодической функцией по оси абсцисс”. Период равен $2\pi r$ (почему — подумайте самостоятельно). Давайте получим два участка кривой (на *рис. 1* представлен один участок). Это значит, что при $r = 1$ значения параметра t должны меняться от 0 до 4π .

Применим для построения циклоиды электронную таблицу (Microsoft Excel или подобную программу). С учетом только что сделанных рассуждений лист должен быть оформлен так (необходимые формулы на основе параметрических уравнений кривой запишите самостоятельно):

	A	B	C	...	S	T
1	t	x	y		r =	1
2	0	0,00000	0,00000			
3	0,05	0,00002	0,00125			
4	0,1	0,00017	0,00500			
5	0,15	0,00056	0,01123			
...						
251	12,45	12,56611	0,00676			
252	12,5	12,56632	0,00220			
253	12,55	12,56637	0,00013			

Рис. 3

По полученным расчетным данным можно построить график (тип диаграммы — **Точечная с гладкими кривыми**). При этом следует учесть, что на диаграммах и графиках электронных таблиц масштаб по осям x и y в общем случае не совпадает, в результате чего вид полученной линии будет отличаться от показанного на *рис. 1*. Сделать масштаб одинаковым можно, меняя размеры области диаграммы.

Лист в формулами и графиком присылайте в редакцию. Все приславшие правильные результаты будут награждены дипломами.

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

Что такое массив

Массив объединяет однотипные элементы программы в наборы — в некоторых языках такие наборы называются коллекциями.

Получается, что массив — это объединение нескольких элементов одного типа. Это могут быть строки, целые числа, двоичные значения.

В разных языках и их разных версиях массивы различаются внешним видом, принципами работы и даже названиями. Например, в Python или JavaScript под массивом чаще всего понимают списки. Вместе с этим списки — это одна из коллекций элементов. Поэтому однозначного строго определения массива нет.

Для лучшего понимания мы разберём несколько вариантов массивов.

Простой пример из жизни

Классический массив можно описать так. Представьте, что у вас есть сумка-холодильник для алюминиевых банок. Она сделана для вещей определённого размера, и у такой сумки есть конечный объём: в неё нельзя положить больше банок, чем предусмотрено.



Обычный массив — это такая же сумка-холодильник. Вы выбираете, какого объёма он будет и что в нём хранить. Как и как сумку-холодильник после покупки, такой массив нельзя изменить после создания — берёшь таким, какой он есть.

Но это только про простые массивы. А ещё есть динамические, двухмерные, многомерные. Про них мы тоже расскажем.

Ключевые термины: элементы, индекс, длина

Главная польза от массива, назовём его хранилищем, появляется при работе с элементами. Значения массивов объединяет их тип. Поэтому можно сказать, что массивом будет не только сумка, но и любой контейнер, в котором хранится один вид вещей: одежда, книги, инструменты, игрушки.

Ещё у каждого элемента есть порядковое число, которое называется индексом. В программировании индексы особенно удобны, потому что сильно влияют на скорость работы.

Длина массива — это его вместимость. Если вы создали массив с длиной 10, в него влезет максимум 10 элементов. Так же в сумку вместимостью 8 банок поместится не больше 8 банок. При этом заполнять массив до максимума необязательно.

Массив в программировании

Массив в программировании — это тоже контейнер, только цифровой.

В разных технологиях массивы различаются. Для примера представим визуально, как выглядит массив из целых чисел и длиной, равной 6. Тут есть элементы и их индексы:



Длина массива равна 6 — это максимальное количество чисел, которое в него влезет:

Обратите внимание, что номер первого элемента — 0. Так происходит потому, что программа присваивает элементам адреса ячеек памяти, где они хранятся. Адрес каждого элемента считается через смещение от первого элемента. У первого элемента смещения нет, поэтому его индекс равен 0.

Для чего нужны массивы в коде?

Массивы могут дать программе несколько преимуществ. Вот несколько основных.

1. **В массиве удобно хранить много значений.** Когда программист объявляет массив, он придумывает название переменной, и в этой переменной могут храниться сотни элементов. При работе мы используем одну переменную и просим выдать значение по нужному индексу. Это экономит память и упрощает структуру программы.
2. **Индексы делают работу быстрее.** Компьютер сразу понимает, где искать нужный элемент, когда мы запрашиваем его по номеру. Поэтому для получения элемента системе всегда нужно одно действие — в разработке такой показатель называется $O(1)$, то есть постоянное время.
3. **Массивы можно быстро обрабатывать,** потому что у них есть такое свойство, как итерируемость. Это значит, что у разработчика есть возможность перебирать все элементы массива и применять к ним какие-то действия. Например, использовать элементы для других вычислений. Некоторые массивы можно изменять, но это зависит от типа массива и языка программирования.
4. **Возможность работы с алгоритмами.** Алгоритмы — один из основных разделов в программировании. Это набор действий, которые решают определённую задачу оптимальным способом. Они могут упорядочивать, представлять наборы значений удобным способом, переходить на другой уровень задач программы. Для большинства алгоритмов нужны массивы.

Сам по себе массив — ещё не залог хорошей работы приложения или сервиса. Но он даёт возможности, которые опытный разработчик может использовать в правильных местах и сильно поднять производительность и функциональность программы.

Как объявить массив

Принцип объявления массива всегда один: нужно дать ему имя и указать, что в нём будет храниться.

Это действие будет выглядеть по-разному для разных языков программирования, поэтому дальше мы посмотрим на несколько вариантов. Обратите внимание, что чаще всего сразу наполнять массив элементами необязательно, но мы заполним, потому что так нагляднее.

Сначала — массив в Python. Обычно под этим словом здесь понимают список, который немного отличается от классического понимания массива: он может содержать элементы любого типа и в любом количестве.

```
# создаём массив с математическими константами
constants = [3.14159, 2.71828, 299792458, 6.67430e-11]
```

Такие же массивы используются в языке JavaScript. Перед объявлением массива нужно поставить тип переменной. Здесь это `let`:

```
// создаём массив с двоичными значениями
let binary = [0b1010, 0b1111, 0b0101, 0b0011];
```

В C++ при создании нужно указать размер массива, чтобы система знала, сколько памяти выделить. Но если мы сразу объявляем элементы, программа поймёт размер автоматически, поэтому нужно указать только тип значений.

Так можно объявить массив булевых (логических) переменных:

```
// создаём массив булевых значений
bool flags[] = {true, false, true, true, false};
```

Выглядят массивы по-разному, но суть везде одна: есть набор значений, которые хранятся в переменной и имеют свой отдельный номер-индекс.

Виды массивов: от простого к сложному

Разберём типы массивов и выясним, почему в языках они могут работать и называться по-разному.

Одномерные массивы — векторы

В традиционной математике «вектор» — набор пронумерованных последовательных значений, записанных в одну строку. Это полностью совпадает с тем, как выглядит самый простой однотипный массив.

Массив-список в Python:

```
numbers = [10, 20, 30, 40]
```

Каждому элементу в таком массиве соответствует один индекс-номер. Чтобы достать нужный элемент, мы пишем название массива и указываем индекс. Чтобы взять 1-й элемент, пишем так:

```
numbers[0]
```

Многомерные массивы — матрицы

Многомерный массив выглядит как таблица-матрица, и иногда его так и называют.

Ещё такие массивы называют вложенными из-за реализации в коде: для его создания внутри массива создаются ещё несколько. Каждый массив внутри будет отдельной строкой, поэтому у всех строк будут свои индексы, а у элементов внутри строк — свои.

Звучит непонятно, поэтому вот наглядный пример матрицы в Python:

```
matrix = [
    [1, 2, 3],
    [4, 5, 6],
    [7, 8, 9]
]
```

Здесь есть три строки:

- Первая: 1, 2, 3.
- Вторая: 4, 5, 6.
- Третья: 7, 8, 9.

Для использования нужного элемента сначала нужно указать номер строки, а потом — номер внутри строки. Для получения числа 5 нужно написать:

```
matrix[1][1]
```

Это будет означать 2-ю строку, 2-й столбец.

Статический массив

Статичность — свойство классического массива, который нельзя изменить после создания.

Списки в Python и других популярных универсальных языках не подходят под это определение, потому что их можно менять: удалять, добавлять и заменять элементы.

Поэтому вот пример статического массива на C++:

```
int values[5] = {1, 2, 3, 4, 5};
```

Эти числа можно брать для работы в других частях программы, но как-то изменить этот набор значений невозможно.

Ещё здесь сразу видно, что работать в C++ немного сложнее, чем в Python: нужно указать тип значений и длину массива. С другой стороны, это сразу заставляет думать заранее о целях своего приложения и делает работу программы быстрее.

Динамический массив

Это массив, который может менять свою длину во время работы программы.

Динамический массив удобнее, но немного медленнее статического. Ещё работать с ним часто гораздо сложнее, если разработчик пишет код на языке, в котором существуют статические массивы и строгое разделение данных по типам.

Современный способ создания динамического массива в C++ выглядит так:

```
// подключаем дополнительные библиотеки vector и iostream
#include <vector>
#include <iostream>
int main() {
    // создаём динамический массив сразу с начальными значениями
    std::vector<int> arr = {10, 20, 30};
    // добавляем новые элементы (вектор автоматически увеличивает
размер)
    arr.push_back(40);
    arr.push_back(50);
    // exit main — память автоматически освобождается
    return 0;
}
```

При этом динамический массив всё равно может хранить только один тип элементов. Поэтому нельзя сказать, что он переменный или гибкий.

Почему все массивы разные

Из-за того, что языки создавались с разными целями, главная идея их реализации сказалась на всей технологии в целом.

Некоторые языки нужны для максимальной скорости и производительности: C, C++, Rust. В этих технологиях работать сложнее, но зато всё находится под контролем разработчика, и можно написать по-настоящему производительную программу. У других языков, таких как Python, корень философии — в удобстве и скорости работы именно программиста, но не самой программы. Поэтому там многое уже сделано за разработчика, но при этом накладывает ограничения на финальное приложение.

Программы для перевода чисел в системах счисления с основаниями, равными степени двойки

В данной статье опишем методики разработки программ для перевода целых чисел из системы в систему, в случаях, когда их основания равны 2, 4, 8 и 16. Как отмечалось в статье в рубрике «Семинар», в таких случаях промежуточный перевод исходных чисел в десятичную систему можно не проводить.

Сначала рассмотрим варианты, когда основание исходной системы счисления меньше.

Программы для таких вариантов аналогичны, поэтому приведем в качестве примера программу для перевода двоичного числа в восьмеричное.

Обозначим исходное число — *число*, а значение, которое нужно получить, — *n*. Обе эти величины будем рассматривать как величины строкового типа.

Пусть, например, *число* = «11110000101». Согласно методике, для перевода нужно, начиная справа, разбить его цифры на триады — группы из трех цифр (110 000 101), и для каждой группы двоичных цифр получить соответствующее десятичное число (восьмеричную цифру). Каждую новую цифру приписать в начало «старого» значения величины *n* (получится значение «605»).

В рассматриваемом случае останутся «необработанными» две цифры исходного числа («11»). Для них также надо найти соответствующую восьмеричную цифру («3») и также добавить ее к «старому» значению *n*. Итак, результат будет равен «3605»

В общем виде схема действий по переводу значений для вариантов, о которых шла речь выше, такая:

1. Обработать все полные группы цифр исходного числа, начиная с его конца, и для каждой из них найти соответствующую цифру «новой» системы. Размер таких групп зависит от оснований систем счисления.
2. Каждую новую цифру приписать в начало «старого» значения величины *n*.
3. Если в исходном числе остались «необработанными» цифры, то для них найти соответствующую цифру «новой» системы и также приписать ее в начало величины *n*.

В программе на школьном алгоритмическом языке, кроме величин *число* и *n*, используем также следующие величины:

- *триада* — группа из трех цифр исходного числа;
- *цифра* — восьмеричная цифра (см. выше);
- *неполн* — оставшиеся «необработанными» цифры (одна или две).

Для разбиения на группы из трех цифр и для контроля того факта, что все полные группы обработаны, применим также величину *послед* — номер последней еще не «обработанной» цифры исходного числа. Начальное значение этой величины равно общему количеству числу цифр в значении *число*, в ходе обработки оно каждый раз уменьшается на 3, а условие для обработки всех полных групп:

послед \geq 3

Для «перевода» последовательности двоичных цифр в восьмеричную цифру целесообразно использовать оператор варианта.

Вся программа имеет вид:

```
алг Перевод_чисел_из_двоичной_системы_в_восьмеричную
нач лит число, n, триада, неполн, сим цифра, цел послед
вывод нс, "Введите исходное число "
ввод число
послед := длин(число)
```

```

n := "" |Первоначальное значение
|Обрабатываем триады
нц пока послед >= 3
    |Выделяем 3 очередные цифры
    триада := число[послед - 2 : послед]
    |Находим соответствующую восьмеричную цифру
    выбор
        при триада = "000": цифра := "0"
        при триада = "001": цифра := "1"
        при триада = "010": цифра := "2"
        при триада = "011": цифра := "3"
        при триада = "100": цифра := "4"
        при триада = "101": цифра := "5"
        при триада = "110": цифра := "6"
        при триада = "111": цифра := "7"
    все
    |и добавляем ее в начало "старого" значения n
    n := цифра + n
    |Изменяем значение послед
    послед :=послед - 3
кц
|Обрабатываем возможные 1-2 цифры
если послед > 0
    то
        неполн := число[1 : послед]
        |Находим соответствующую восьмеричную цифру
        выбор
            при неполн = "1": цифра := "1"
            при неполн = "10": цифра := "2"
            при неполн = "11": цифра := "3"
        все
        |и учитываем ее
        n := цифра + n
все
вывод нс, "Полученное число: ", n
кон

```

Задание для самостоятельной работы

Разработайте программы для перевода целого числа:

- 1) из двоичной системы в четверичную;
- 2) из двоичной системы в шестнадцатеричную;
- 3) из четверичной системы в шестнадцатеричную.

Теперь обсудим рассмотрим варианты программ, когда основание системы счисления, в которой представлено исходное число, больше, и в качестве примера — перевод восьмеричного числа в двоичное.

Обозначим в программе исходное число — n , а значение, которое нужно получить, — *число* (обе величины будут строкового типа).

Согласно методике (см. статью в рубрике «Семинар»), каждую цифру заданного числа, рассматривая её как десятичную, нужно представить в виде трёхразрядного двоичного числа (может быть — с начальным нулем, с начальными нулями или состоящим из трех нулей) и из полученных двоичных чисел сформировать единое число (без возможных начальных нулей). Это означает, что в программе первая слева цифра должна быть обработана отдельно. Учитывая это, в программе используем также следующие величины:

- *цифра* — отдельные цифры исходного восьмеричного числа;
- *триада* — группа из трех двоичных цифр;

— *перв* — последовательность двоичных цифр, соответствующая первой цифре исходного числа.

Программа:

```
алг Перевод_чисел_из_восьмеричной_системы_в_двоичную
нач лит n, число, триада, перв, сим цифра, цел послед
вывод нс, "Введите исходное число "
ввод n
число := "" |Первоначальное значение
|Обрабатываем все символы величины n, кроме 1-го
нц для i от длин(n) до 2 шаг -1
    |Выделяем каждую цифру
    цифра := n[i]
    |Находим соответствующую триаду
    выбор
        при цифра = "0": триада := "000"
        при цифра = "1": триада := "001"
        при цифра = "2": триада := "010"
        при цифра = "3": триада := "011"
        при цифра = "4": триада := "100"
        при цифра = "5": триада := "101"
        при цифра = "6": триада := "110"
        при цифра = "7": триада := "111"
    все
    |и добавляем ее в начало "старого" значения
    |величины число
    число := триада + число
кц
|Обрабатываем 1-ю цифру
|Определяем ее
цифра := n[1]
|Находим соответствующее двоичное представление
выбор
    при цифра = "1": перв := "1"
    при цифра = "2": перв := "10"
    при цифра = "3": перв := "11"
    при цифра = "4": перв := "100"
    при цифра = "5": перв := "101"
    при цифра = "6": перв := "110"
    при цифра = "7": перв := "111"
все
|и также добавляем его
число := перв + число
вывод нс, "Полученное число: ",число
кон
```

Задание для самостоятельной работы

Разработайте программы для перевода целого числа:

- 1) из четверичной системы в двоичную;
- 2) из шестнадцатеричной системы в двоичную;
- 3) из шестнадцатеричной системы в четверичную.

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 Пойя

Друзья и удочки

Пять друзей – Даниил, Саркис, Курбан, Семён и Юлиан – решили купить себе удочки. В магазине были удочки пяти цветов: белого, черного, красного, синего и зелёного.

Известно, что:

- Даниил любит красный и синий цвета;
- Семену нравится синяя и зелёная удочки;
- Курбан купил зелёную удочку;
- Юлиан отдавал предпочтение красной, синей и черной удочкам.

Кто какую удочку купил, если у всех оказались удочки разного цвета?

Из Елкино — в Палкино

Между четырьмя местными аэропортами: Елкино, Палкино, Веревкино и Булкино ежедневно выполняются авиарейсы. Расписание перелетов рейсов между ними:

Аэропорт вылета	Аэропорт прилета	Время вылета	Время прилета
Елкино	Веревкино	7:30	10:50
Булкино	Елкино	8:15	10:35
Палкино	Веревкино	11:35	13:25
Веревкино	Елкино	12:10	14:20
Елкино	Булкино	12:30	14:30
Булкино	Палкино	14:10	16:20
Елкино	Палкино	14:15	16:40
Веревкино	Палкино	14:20	16:30
Палкино	Елкино	16:10	18:50
Палкино	Булкино	18:40	20:45

Представьте, что вы оказались в аэропорту Елкино в 6 часов утра. Каково самое раннее время, когда вы сможете попасть в аэропорт Палкино? Время на пересадку из самолета в самолет не учитывать.

Серебряная цепочка

В гостиницу приехал путешественник. Выяснилось, что деньги он забыл дома. У него была лишь серебряная цепочка из 7 звеньев. Он договорился с хозяином гостиницы о том, что за каждый день пребывания в гостинице будет расплачиваться одним звеном цепочки. Какое звено цепочки надо аккуратно расцепить, чтобы прожить в гостинице семь дней и ежедневно расплачиваться с хозяином? (Хозяин может давать сдачу звеньями, полученными им ранее.) Как потом он сможет расплачиваться?

Помогите Марии Петровне

Мария Петровна рассказала о своем разговоре с соседкой – бабушкой Пелагеей:

— Мои четыре внучки — замечательные девочки, — рассказывала бабушка Пелагея с нескрываемой гордостью. — Каждая из них играет на каком-нибудь музыкальном инструменте и говорит на одном из иностранных языков.

— На чем играет Маша? — спросила я.

— На рояле.

— А кто играет на скрипке?

— Помню только, что это та девочка, которая говорит по-французски, — ответила бабушка.

Поговорив с бабушкой, я также узнала, что Оля играет на виолончели, а Лена не говорит по-немецки. Маша не знает итальянского языка, а Оля не владеет английским. Валя не знает французского, Лена не играет на арфе, а виолончелистка не говорит по-итальянски.

После такого рассказа Мария Петровна запуталась. Помогите ей определить, кто на каком инструменте играет и на каком иностранном языке говорит.

Задача “Равенства”

В какой системе счисления справедливы равенства:

а) $3 \times 4 = 10$;

б) $5 \times 5 = 31$;

в) $2 \times 2 = 100$.

Кто есть кто?

Кто из двух людей — А и В — рыцарь (всегда говорит правду), а кто лжец (всегда врет), если А сказал: “По крайней мере, один из нас лжец”.

Крестики–нолики

Трое друзей (врач, учитель и маркетолог) играли в «крестики-нолики». Каждый сыграл с каждым один раз. Константин взял верх над врачом. Андрей обыграл маркетолога. А встреча Ивана и Константина завершилась вничью. Как звали учителя?

Пропавший доллар

Трое путешественников прибыли на постоялый двор, где решили заночевать. Свободной оказалась лишь одна комната с тремя кроватями, за которую хозяин запросил 30 долларов. Поэтому каждый путешественник заплатил по 10 долларов.

Позднее, подсчитывая выручку, хозяин вспомнил, что за комнату следовало взять не 30, а только 25 долларов. Он послал мальчика отнести пять монет по одному доллару путешественникам. Но мальчик по ходу дела взял свои чаевые и вручил каждому путешественнику по одному доллару, оставив себе два доллара.

Итак, каждый путешественник заплатил за ночлег по 9 долларов, а два доллара мальчик оставил себе. Но трижды девять плюс два – это 29. Куда же делся один доллар?

«Электросчётное пианино», предиктограф и машина логики: какой писатели-фантасты видели вычислительную технику будущего. Продолжение

А.И. Басов, Москва
В.В. Шилов, Москва

Литература быстро заметила появление новых технологий. В 1908 году ученый и писатель Александр Александрович Богданов публикует роман «Красная звезда». Марсиане приглашают главного героя — русского революционера — посетить Марс, где уже давно построен коммунизм. До Марса герои летят на космическом корабле с ядерным двигателем (одно из предвидений писателя), на борту корабля имеется вычислительная машина для расчета траектории полета. Вот как описывает ее Богданов: «Далее шла “вычислительная” комната. Там стояли непонятные для меня машины со множеством циферблатов и стрелок. За самой большой машиной работал Стэрни. Из нее тянулась длинная лента, заключавшая, очевидно, результаты вычислений...».

Именно так выглядели первые компьютеры: просторные залы, заполненные шкафами с оборудованием, автоматическая пишущая машинка для вывода результатов... Упоминание множества циферблатов и стрелок как раз и намекает на то, что Богданов видел табуляторы Германа Холлерита, в которых информация отображалась при помощи похожих на часы счетчиков (рис. 4).



Рис. 4. Циферблаты табулятора Германа Холлерита
(из коллекции Политехнического музея)

В то же время марсианские вычислительные машины используются только во время космических полетов, о других применениях в романе не говорится. Таким образом, Богданов предсказал появление бортовых компьютеров — специализированных, а не универсальных вычислительных машин. Прилетев на Марс, главный герой обнаруживает высокотехнологичное общество, где есть 3D-кино и телевидение, автоматические станки, синтетические ткани. Так как все производства принадлежат государству, рынка труда нет. Его заменяет бюро статистики, которое собирает информацию об избытке и нехватке трудовых ресурсов в разных отраслях. Для этого используется особая система:

«...черные стены были покрыты рядами блестящих белых знаков — это были просто таблицы статистики труда... В то время как мы таким образом разговаривали, я вдруг заметил, что некоторые цифры таблицы исчезли, а затем на их месте появились новые. Я спросил, что это значит.

— Цифры меняются каждый час ... в течение часа несколько тысяч человек успели заявить о своем желании перейти с одних работ на другие. Центральный статистический механизм все время отмечает это, и каждый час электрическая передача разносит его сообщения повсюду».

На рис. 5 справа мы видим «стены с белыми знаками», какими их представил себе художник-иллюстратор. А на этом же рисунке слева на столе в кабинете марсианина стоит пишущая машинка, преобразующая речь в печатный текст. Марсианский «Центральный статистический механизм», действующий в масштабах всей планеты — это прообраз современных глобальных информационных систем. Что, в свою очередь, предполагает существование некой «всемарсианской сети» передачи информации (Богданов уточняет, что эта сеть «электрическая»). Однако, разумеется, это не наш сегодняшний интернет, ведь система в романе Богданова предназначена только для одного вида данных, т.е. она специализированная.



Рис. 5. Слева — пишущая машинка, преобразующая речь в текст. Справа — «табло» марсианской автоматической информационной системы (иллюстрации к роману «Красная звезда», художник И.А. Бабий, 1922 год)

Интересно, что Богданов пишет о том, что марсианская статистическая система требует огромных до статочной точности вычислений, но при этом не упоминает об использовании для этого каких-либо счетных машин...

Роман Богданова неоднократно издавался и хорошо известен, но почти такая же система была описана за 12 лет до его выхода. В романе «На другой планете» писателя Порфирия Павловича Инфантьева главный герой тоже попадает на Марс, где построен коммунизм. Описывает Инфантьев и «Главное Центральное Статистическое Бюро», в котором находятся «огромные; богато изукрашенные шкафы с циферблатами, расставленные по всей длине громадной залы». Циферблаты, очевидно, тоже показывают цифры, связанные с плановым ведением разных отраслей промышленности и сельского хозяйства. Так, наверху одного из «огромных шкафов» действительно «красовалась надпись: “Агрикультура”». К сожалению, роман Инфантьева перед публикацией подвергся цензуре, и глава, в которой могло находиться более подробное описание организации статистического бюро и находящихся в нем машин, была изъята...

Девять ребусов на одну тему

Ребус 1



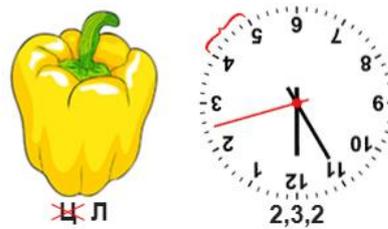
Ребус 2



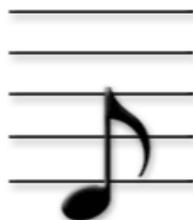
Ребус 3



Ребус 4



Ребус 5



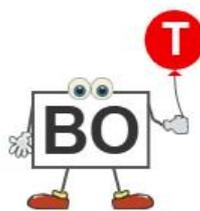
Н+Е

Ребус 6



Ребус 7

ОЛ+И



Ребус 8



1 = Б

Ребус 9



3 = ЙТ

Ответы присылайте в редакцию. Можно решать не все ребусы.

Головоломка “Последняя табличка”

Определите, как должна выглядеть последняя табличка, чтобы закон изменения положения символов во всех табличках сохранился.

7	X					X					
6											
5				X			○	#			
4				#						#	
3					○				X		
2		○	#			X					
1											

На шахматной доске”

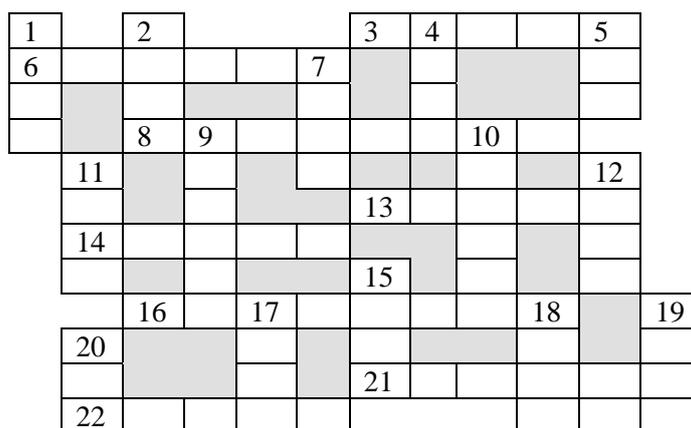
Перечислите все маршруты, ведущие из клетки c3 шахматной доски в клетку a1, при соблюдении следующего правила: каждый ход делается либо на одну клетку влево, либо на одну клетку вниз, либо на одну клетку вниз и одну клетку влево.

Положение на шахматной доске клеток a1 и c3 установите самостоятельно.



Кроссворд

Решите, пожалуйста, кроссворд.



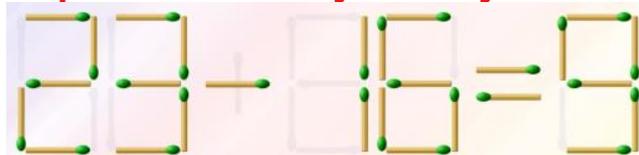
По горизонтали

3. Один из контактов транзистора типа **МОП** (Металл — Оксид — Полупроводник).
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.

Переложить одну спичку и сделать равенство верным

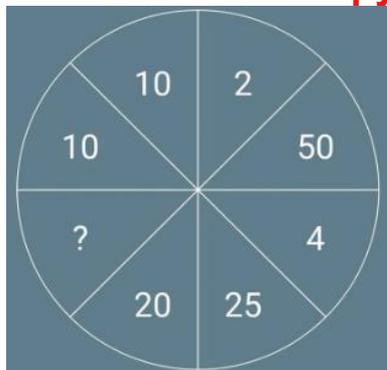


Задача предназначена для учеников 1–7-х классов.

Таблетки согласно назначению врача

Врач назначил одному человеку принимать «Антиболин» и «Выздоровин» по одной таблетке утром и вечером. При этом нельзя выпить две таблетки одного вида утром, а другие две – вечером. Он приготовил четыре таблетки на день, но они перемешались и выглядят одинаково. Как ему выпить таблетки согласно назначению врача?

Восемь чисел в круге



Какое число должно быть записано вместо вопросительного знака.

Крест — накрест

Переставив буквы в строках приведённого ниже квадрата, получите слова, при этом в диагоналях квадрата соберутся еще два слова, связанные с информатикой и компьютерами, а также название ягоды, которую любят все программисты. Найдите все слова и дайте комментарии к ним.

П	А	С	И	З	Ь
Н	Е	П	А	Л	Ь
М	О	Н	Т	Е	Р
В	Е	Д	Р	О	Е
Н	А	Л	И	М	А
К	А	В	Р	А	П

Квадратики и цифры

В квадратике впишите по одной цифре так, чтобы выполнялись указанные действия. При этом цифра "ноль" в ребусе отсутствует.

$$\begin{array}{ccc} \square & : & \square = \square \\ - & \times & \times \\ \square & + & \square = \square \\ = & = & = \\ \square & + & \square = \square \end{array}$$

Две задачи на переливание

1. Есть две ведра А и В емкостью 3 л и 10 л, соответственно. Как отмерить 5 л?
2. Есть три ведра А, В и С емкостью 3 л, 5 л и 10 л, соответственно. Как отмерить 9 л?

Эллипсоид

Разложим слово ЭЛЛИПСОИД на слоги:

ЭЛ ЛИП СО ИД

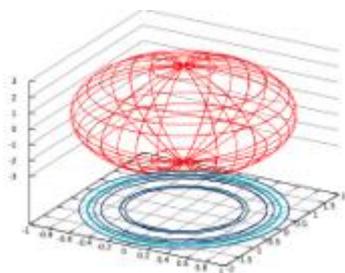
Замените буквы цифрами так, чтобы получившиеся четыре числа были точными квадратами. При этом одинаковым буквам соответствуют одинаковые цифры, а разным - разные.

Рекомендации по решению

Обратите внимание на двузначные числа. В них нет двух одинаковых цифр. Это значит, что среди них нет точных квадратов с одинаковыми цифрами, например, 16 и 36, 49 и 64 и др.

Выпишите все тройки двузначных чисел, являющихся точными квадратами и исключите из те, в которых есть одинаковые цифры. Останутся 6 троек чисел.

Далее определите, какой может быть вторая буква Л, являющая первой в трехзначном числе. Список трехзначных точных квадратов найдите в Интернете.



Эллипсоид — поверхность в трёхмерном пространстве, полученная деформацией сферы (шара) вдоль трёх взаимно перпендикулярных осей

Задача про автомат (но не Калашникова)

Автомат получает на вход четырехзначное число. По этому числу строится новое число по следующим правилам:

- 1) складывается первая и вторая, а также третья и четвертая цифры исходного числа;
- 2) полученные два числа записываются друг за другом в порядке убывания (без разделителей).

Пример. Исходное число: 5348. Суммы: $5 + 3 = 8$, $4 + 8 = 12$.

Укажите наименьшее число, в результате обработки которого автомат выдаст число 1412.

Дедушка и внук



Дедушке и внуку вместе 77 лет. Возраст дедушки равен возрасту внука, если прочитать последний «справа налево». Сколько лет каждому?

Задача предназначена для учеников 1–7-х классов.

Конкурс № 77 «Графы». Тур 3 (заключительный)

Напомним, что в качестве задания этого конкурса, который будет проходить в несколько туров, предложены вопросы, связанных с так называемыми «графами». Итоги конкурс будут подводиться с учетом всех туров в целом.

Вопрос 5

Сколько различных неориентированных² графов-семиугольников (в том числе невыпуклых) можно получить, соединив семь точек из девяти? Какие? (Изобразите их в ответе).

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 марта 2026 г.

Ответы и решения задач и головоломок, опубликованных в декабрьском номере нашего журнала, представили:

— Апраксина Елена (4), Жук Марина (6) и Яркина Надежда (3), средняя школа села Муравьёво Вологодской обл., учитель **Муравьёва О.В.**;

— Аتماжар Ирина (7), Борисова Дарья (2), Замилацкая София (3), Короткин Егор (7), Порохин Данил (2), Свиначев Денис (4), Семенова Ирина (5), Степанян Габриэль (5), Степанян Даниэль (7) и Юсюфзай Аджира (4), ученики 4-го класса школа № 11 г. Струнино Владимирской обл., учитель **Волкова Т.П.**;

— Безруких Елена (4) и Щукина Татьяна (5), средняя школа поселка Осиновка, Алтайский край, учитель **Евдокимова А.И.**;

— Беликов Максим (8), Куценко Сергей (8) и Шмарова Ирина (5), г. Пенза, школа № 73, учитель **Диков А.В.**;

— Бурцев Николай (5), Бурчак Константин (6), Грезин Михаил (4), Дауд Григорий (5), Джафарова Мирослава (5), Кулаков Егор (1), Лузянин Антон (1), Мурзина Евгения (7), Новикова Дарья (1) и Саргсян Диана (2). Владимирская обл., г. Струнино, школа № 11, учитель **Волков Ю.П.**;

— Бякина Анна (3), средняя школа села Восточное Нижегородской обл., учитель **Долгова Г.А.**;

— Вагизов Ислам и Рахимов Ильназ, Республика Татарстан, Мамадышский р-н, совхоз «Мамадышский», политехнический колледж, преподаватель **Порываева Н.С.**;

— Василенко Сергей (5), Лось Марина (6) и Яковенко Станислав (3), средняя школа села Обнинское Красноярского края, учитель **Потапов М.И.**;

— Ведерников Михаил, Владимирская обл., г. Александров, школа № 13, учитель **Семенкович А.О.**;

— Карапетян Армен (4) и Лазо Владимир (6), Москва, школа № 1530 «Школа Ломоносова», учитель **Каменецкая Т.С.**;

— Малышева Серафима (6), средняя школа села Горелово Тамбовской обл., учитель **Шитова Л.А.**

В скобках указано количество решенных задач и головоломок (без учета задания конкурса).

Спасибо всем!

² Неориентированный граф – граф, в котором рёбра не имеют определённого направления.

