**ВВЕДЕНИЕ**

Я выбрал эту тему, потому что мне было очень интересно узнать, как компьютер получает информацию, как он производит арифметические действия над числами. Еще мне хотелось узнать, что такое система счисления, ее виды, как считали в разные века, в разных странах люди. Как переводить одну систему счисления в другую.

При решении задач у нас получилось, что мы собрали информацию с разных источников. Далее мы проанализировали всю собранную нами информацию, и выбрали необходимую нам. Системы счисления - знаковые системы, в которых числа записываются по определенным правилам с помощью символов некоторого алфавита, называемых цифрами.

Современный человек в повседневной жизни постоянно сталкивается с числами и цифрами: запоминает номера автобусов и телефонов, в магазине подсчитывает стоимость покупок, ведет свой семейный бюджет в тенге или копейках и т.д. Числа и цифры с нами везде! Интересно, что знал человек о числах две тысячи лет назад? А пять тысяч лет назад?

Историки доказали, что и пять тысяч лет тому назад люди могли записывать числа, могли производить над ними арифметические действия. При этом записывали они числа совершенно по другим принципам, нежели мы в настоящее время.

Понятие числа - фундаментальное понятие, как математики, так и информатики. Под числом мы будем понимать его величину, а не его символьную запись. Сегодня человечество для записи чисел использует в основном десятичную систему счисления. Что же такое - система счисления? Это мы узнаем в ходе изучения материала. Перед собой поставили -следующую цель: узнать, для чего нужна система счисления.

Для достижения поставленной цели сформулировали следующие задачи: изучить литературу о различных системах счисления, почему в ЭВМ информация представляется в двоичной системе счисления и чем она удобна, где еще используется двоичная система счисления.

Причина, по которой десятичная система счисления стала общепринятой, вовсе не математическая. Десять пальцев рук - вот аппарат для счета, которым человек пользуется с доисторических времен. Древнее написание десятичных цифр (см. рис. 1):



Рис. 1 - Написание цифр в древности

Довольно широкое распространение имела двенадцатеричная система счисления. Происхождение ее тоже связано со счетом на пальцах. Считали большой палец руки и фаланги остальных четырех пальцев: всего их 12 (см. Приложение, рис. 1).

Особо важную роль играл природный инструмент человека – его пальцы. Этот инструмент не мог длительно хранить результат счета, но зато всегда был «под рукой» и отличался большой подвижностью. Язык первобытного человека был беден; жесты возмещали недостаток слов, и числа, для которых еще не было названий, «показывались» на пальцах.

**История возникновения системы счисления**

Первобытному человеку считать почти не приходилось. "Один", "два" и "много" - вот все его числа. Современным людям приходится иметь дело с числами буквально на каждом шагу. Нужно уметь правильно назвать и записать любое число, как бы велико оно ни было. Если бы каждое число называлось особым именем и обозначалось в письме особым знаком, то запомнить все эти слова и знаки было бы никому не под силу. Как же справиться с этой задачей? Нас выручает хорошая система обозначений.

Однако не во все времена и не везде люди пользовались десятичной системой. С точки зрения чисто математической она не имеет специальных преимуществ перед другими системами счисления, и своим повсеместным распространением эта система обязана вовсе не общим законам математики, а причинам совсем иного характера.

В последнее время с десятичной системой серьезно конкурируют двоичная и, отчасти, троичная системы, которыми "предпочитают" пользоваться современные вычислительные машины.

В некоторых странах (например, Китае и Японии) иероглифическое письмо сохранилось и до наших дней.

Унарные системы счисления

Унарные системы счисления - системы счисления, алфавит которых содержит всего один символ. Число образуется путем повторения одного знака, символизирующего единицу.

В качестве единственной «[цифры](http://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%B0)» используется «1», чёрточка (|), камешек, костяшка [счёт](http://ru.wikipedia.org/wiki/%D0%A1%D1%87%D1%91%D1%82%D1%8B), узелок, зарубка и др.

Полинарные системы счисления - системы счисления, алфавиты которых содержат много символов.

По зависимости значения цифры от её местоположения в числе они бывают:

- Непозиционные системы счисления;

- Мультипликативные системы счисления;

- Позиционные системы счисления.

Непозиционные системы счисления

Непозиционные системы счисления – полинарные системы счисления, в которых знаку, представляющему собой цифру, всегда соответствует определенное значение, вне зависимости от его место положения в записи числа.

Как только люди начали считать, у них появилась потребность в записи чисел. Находки археологов на стоянках первобытных людей свидетельствуют о том, что первоначально количество предметов отображали равным количеством каких-либо значков (бирок): зарубок, черточек, точек.

Римскими цифрами пользовались очень долго. Еще 200 лет назад в деловых бумагах числа должны были обозначаться римскими цифрами (считалось, что обычные арабские цифры легко подделать). Римская система счисления сегодня используется, в основном, для наименования знаменательных дат, томов, разделов и глав в книгах.

Мультипликативные системы счисления -полинарные системы счисления, в которых для записи чисел используется уже определённое количество цифр, которые могут принимать разные значения в зависимости от расположения в записи числа.

В позиционных системах счисления – количественный эквивалент каждой цифры зависит от ее положения (позиции) в коде(записи) числа. Ныне мы привыкли пользоваться десятичной позиционной системой — числа записываются с помощью 10 цифр. Самая правая цифра обозначает единицы, левее — десятки, ещё левее — сотни и т.д.

Двоичная система счисления — [позиционная система счисления](https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B7%D0%B8%D1%86%D0%B8%D0%BE%D0%BD%D0%BD%D0%B0%D1%8F_%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0_%D1%81%D1%87%D0%B8%D1%81%D0%BB%D0%B5%D0%BD%D0%B8%D1%8F) с основанием Благодаря непосредственной реализации в [цифровых электронных схемах](https://ru.wikipedia.org/wiki/%D0%A6%D0%B8%D1%84%D1%80%D0%BE%D0%B2%D0%B0%D1%8F_%D1%8D%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%B8%D0%BA%D0%B0) на [логических вентилях](https://ru.wikipedia.org/wiki/%D0%9B%D0%BE%D0%B3%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B8%D0%B9_%D0%B2%D0%B5%D0%BD%D1%82%D0%B8%D0%BB%D1%8C), двоичная система используется практически во всех современных [компьютерах](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80) и прочих вычислительных [электронных устройствах](https://ru.wikipedia.org/wiki/%D0%AD%D0%BB%D0%B5%D0%BA%D1%82%D1%80%D0%BE%D0%BD%D0%BD%D0%BE%D0%B5_%D1%83%D1%81%D1%82%D1%80%D0%BE%D0%B9%D1%81%D1%82%D0%B2%D0%BE).Таким образом, в современных информационных технологиях при создании программного обеспечения в основном используется двоичная система счисления, так как компьютеру легче оперировать большим числом простых элементов, чем небольшим числом сложных.

**Арифметические операции в различных системах счисления**

Сложение и вычитание

В системе с основанием для обозначения нуля и первых ρ-1 натуральных чисел служат цифры 0, 1, 2, ..., ρ - 1. Для выполнения операции сложения и вычитания составляется таблица сложения однозначных чисел.

Таблица 1 - Сложение в двоичной системе

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| + | 0 | 1 | 2 |  |  |  | q-1 |
| 0 | 0 | 1 | 2 | \*\*\* | \*\*\* | \*\*\* | q-1 |
| 1 | 1 | 2 | 3 | \*\*\* | \*\*\* | \*\*\* | 10 |
| 2 | 2 | 3 | 4 | \*\*\* | \*\*\* | \*\*\* | 11 |
| \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* | \*\*\* |
| q-1 | q-1 | 10 | 11 | \*\*\* | \*\*\* | \*\*\* | 1(q-2) |

Например, таблица сложения в шестеричной системе счисления:

Таблица 2 - Сложение в шестеричной системе

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| + | 0 | 1 | 2 | 3 | 4 | 5 |
|  |  |  |  |  |  |  |
| 0 | 0 | 1 | 2 | 3 | 4 | 5 |
| 1 | 1 | 2 | 3 | 4 | 5 | 10 |
| 2 | 2 | 3 | 4 | 5 | 10 | 11 |
| 3 | 3 | 4 | 5 | 10 | 11 | 12 |
| 4 | 4 | 5 | 10 | 11 | 12 | 13 |
| 5 | 5 | 10 | 11 | 12 | 13 | 14 |

Сложение любых двух чисел, записанных в системе счисления с основанием ρ, производится так же, как в десятичной системе, по разрядам, начиная с первого разряда, с использованием таблицы сложения данной системы. Складываемые числа подписываются одно за другим так, чтобы цифры одинаковых разрядов стояли по вертикали. Результат сложения пишется под горизонтальной чертой, проведенной ниже слагаемых чисел. Так же как при сложении чисел в десятичной системе, в случае, когда сложение цифр в каком-либо разряде дает число двузначное, в результат пишется последняя цифра этого числа, а первая цифра прибавляется к результату сложения следующего разряда.

Например,



Можно обосновать указанное правило сложения чисел, используя представление чисел в виде:



Разберем один из примеров:

=3\*72+5\*71+4\*70

=2\*72+6\*71+3\*70

Имеем:

(3\*72+5\*71+4\*70) + (2\*72+6\*71+3\*70) =(3+2)\*72+(5+6)\*7+(3+4)=

=5\*72+1\*72+4\*7+7=6\*72+4\*7+7=6\*72+5\*7+0=6507

Последовательно выделяем слагаемые по степени основания 7, начиная с низшей, нулевой, степени.

Вычитание производится также по разрядам, начиная с низшего, причем если цифра уменьшаемого меньше цифры вычитаемого, то из следующего разряда уменьшаемого "занимается" единица и из полученного двузначного числа вычитается соответствующая цифра вычитаемого; при вычитании цифр следующего разряда в этом случае нужно мысленно уменьшить цифру уменьшаемого на единицу, если же эта цифра оказалась нулем (и тогда уменьшение ее невозможно), то следует "занять" единицу из следующего разряда и затем произвести уменьшение на единицу. Специальной таблицы для вычитания составлять не нужно, так как таблица сложения дает результаты вычитания.

Например,

Умножение и деление

Для выполнения действий умножения и деления в системе с основанием ρ составляется таблица умножения однозначных чисел.

Таблица 3 - Умножение однозначных чисел

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| \* | 0 | 1 | 2 |  |  |  | q-1 |
| 0 | 0 |  |  |  |  |  | 0 |
| 1 | 0 |  |  |  |  |  | q-1 |
| 2 | 0 |  |  |  |  |  | 1(q-1) |
| \*\*\* | \*\*\* |  |  |  |  |  | \*\*\* |
| q-1 | 0 |  |  |  |  |  | 1(q-2) |

Таблица 4 - Умножение в шестеричной системе счисления

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| \* | 0 | 1 | 2 | 3 | 4 | 5 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 | 5 |
| 2 | 0 | 2 | 4 | 10 | 12 | 14 |
| 3 | 0 | 3 | 10 | 13 | 20 | 23 |
| 4 | 0 | 4 | 12 | 20 | 24 | 32 |
| 5 | 0 | 5 | 14 | 23 | 32 | 41 |

Умножение двух произвольных чисел в системе с основанием ρ производится так же, как в десятичной системе - "столбиком", то есть множимое умножается на цифру каждого разряда множителя (последовательно) с последующим сложением этих промежуточных результатов.

Например,



При умножении многозначных чисел в промежуточных результатах индекс основания не ставится:



Деление в системах с основанием ρ производится углом, так же, как в десятичной системе счисления. При этом используется таблица умножения и таблица сложения соответствующей системы. Сложнее дело обстоит, если результат деления не является конечной ρ-ичной дробью (или целым числом). Тогда при осуществлении операции деления обычно требуется выделить непериодическую часть дроби и ее период. Умение выполнять операцию деления в ρ-ичной системе счисления полезно при переводе дробных чисел из одной системы счисления в другую.

Например:



**Перевод чисел из одной системы счисления в другую**

Существует много различных способов перевода чисел из одной системы счисления в другую.

Способ деления

Пусть дано число N=an an-1. . . a1 а0 р.

Для получения записи числа N в системе с основанием h следует представить его в виде:

N=bmhm+bm-1hm-1+... +b1h+b0 (1)

где 1<bm<h-1, 0 ≤ bi ≤ h-l (i=0, 1,... ,m-l), тогда

=bmbm-1... b1boh (2)

Из (1) получаем:

= (bmhm-1+...+b)\*h +b0 = N1h+b0, где 0≤ b0 ≤h (3)

есть, цифра b0 является остатком от деления числа N на число h. Неполное частное Nl = bmhm-1+ . . . +b1 представим в виде:

= (bmhm-2 + ... + b2)h + b1 = N2h+b1, где 0≤ b2 ≤h (4)

Таким образом, цифра bi в записи (2) числа N является остатком от деления первого неполного частного N1 на основание h новой системы счисления. Второе неполное частное N2 представим в виде:

= (bmhm-3+ ... +b3)h+b2, где 0≤ b2 ≤h (5)

то есть цифра b2 является остатком от деления второго неполного частного N2 на основание h новой системы. Так как не полные частные убывают, то этот процесс конечен. И тогда мы получаем Nm = bm, где bm<h, так как:

Nm-1 = bmh+bm.1 = Nmh+bm.1

Таким образом, последовательность цифр bm, bm-1 . . ,b1,b0 в записи числа N в системе счисления с основанием h есть последовательность остатков последовательного деления числа N на основание h, взятая в обратной последовательности.

Рассмотрим пример: Выполнить перевод числа 123 в шестнадцатеричную систему счисления:



Таким образом, число 12310=7(11)16 либо можно записать как 7B16

Запишем число 340227 в пятеричной системе счисления:



Таким образом, получаем, что 340227=2333315.

Вывод:

Таким образом, для перевода целого двоичного числа восьмеричное его нужно разбить на группы по три цифры, справа налево, а затем преобразовать каждую группу в восьмеричную цифру. Если в последней, левой, группе окажется меньше трех цифр, то нужно его дополнить нулями слева.

Десятичная система счисления

Представьте, что вы пересчитываете большое число одинаковых предметов, например, спичек. Удобнее всего будет разложить эти предметы как кучки по десять в каждой. Получится некоторое количество десятков (и, может быть, останутся несколько предметов, не вошедших в целые десятки). Далее придется пересчитывать кучки (десятки). Если кучек (десятков) будет очень много, можно их тоже сгруппировать в десятки, и так далее.

Счет десятками получил широкое распространение потому, что люди располагают естественной "счетной машиной", связанной с числом десять -десятью пальцами на руках.

Десятичная нумерация "изобретена" индусами; в Европу ее занесли арабы, вторгшиеся в Испанию в VIII в. нашей эры. Арабская нумерация распространилась по всей Европе, и, будучи проще и удобнее остальных систем счисления, быстро их вытеснила. До сих пор наши цифры принято называть арабскими. Впрочем, за тысячу лет все цифры, кроме 1 и 9, сильно изменились. Вот, для сравнения, наши (называемые арабскими) и настоящие арабские цифры





Двенадцатеричная система счисления - ее происхождение тоже связано со счетом на пальцах: так как четыре пальца руки (кроме большого) имеют в совокупности двенадцать фаланг, то по этим фалангам, перебирая их по очереди большим пальцем, и ведут счет от одного до двенадцати. 

Троичная система счисления– позиционная система счисления с целочисленным основанием равным 3. Она существует в двух вариантах: *несимметричная* и *симметричная*троичные системы. Несимметричная система обычно использует символы: 0, 1 и 2. Симметричная: –1, 0, +1. десятичные числа и соответствующие им числа в троичной системе счисления.

 Элементы троичной системы существовали еще у древних шумеров. Полноценную симметричную троичную систему впервые предложил итальянский математик *Фибоначчи*(*Леонардо Пизанский*) (1170–1250).

Восьмеричная система счисления - позиционная система счисления с основанием 8. Для представления чисел в ней используются арабские цифры. Используется всего восемь цифр - 0, 1, 2, 3, 4, 5, 6, 7. Шестидесятеричная система счисления.

Шестидесятеричная система счисления существовала и возникла в Древнем Вавилоне. Мнения историков по поводу того, как именно возникла эта система, расходятся. Одна из гипотез, состоит в том, что произошло слияние двух племен, одно из которых пользовалось шестеричной системой счисления, а второе - десятичной. Шестидесятеричная система возникла как компромисс между этими двумя системами.

Из перечисленных выше систем счисления, сыгравших, наряду с десятичной, заметную роль в развитии человеческой культуры, все, кроме шестидесятеричной, источники которой не ясны, связаны с тем или иным способом счета по пальцам рук (или и рук, и ног), то есть имеют несомненное "анатомическое" происхождение.

**Применение различных видов систем счисление в жизни.**

1 Деньги в конвертах и зерна на шахматной доске

Поставим перед собой задачу. Допустим, что банкиры, занимающиеся отмыванием грязных денег, и завтра ждут важного клиента, которому должны выдать круглую или не очень круглую, но заранее им известную сумму от 1 до 1 000 000 000$ чтобы не пачкать руки о грязные деньги, они заранее дали своим кассирам заготовить некоторое количество конвертов с деньгами, на которых написаны содержащиеся в них суммы, и собираются просто отдать клиенту один или несколько конвертов, в которых и будет содержаться требуемая им сумма. Какое количество конвертов необходимо иметь?

Конечно, можно просто заготовить конверты со всеми суммами от 1 до 1 000 000 000, но где взять столько денег на конверты?

Узнаем, какова будет в этом случае полная сумма во всех конвертах? Попробуем оценить также массу бумаги, предполагая, что использованы не более чем сотенные купюры.

Есть более рациональные подход к нашему делу. Надо положить в первый конверт 1 $, а в каждый следующий класть вдвое большую сумму, чем в предыдущий. Тогда, например, в 5-м конверте будет 16 $., в 10-м – 512$, в 11-м -1024 у.е., в 21-м -1024 = 1 048 576$., в 31-м -1024 = 1 073 741 824 $., но он, очевидно, уже не понадобится, а вот 30-й с 1 073 741 824/2 = 536 870 912 $ может и пригодиться. В общем случае сумма в (n + 1)-м конверте будет равна произведению n двоек, это число принято обозначать 2 и называть n-й степенью двойки. Условимся считать, что 20 = 1. проведенные выше вычисления основались на следующих свойствах операции возведения в степень: 2n2m = 2n+m, 2n /2m = 2n-m, (2n)m = 2nm

Экспериментально легкое проверить, что любое число можно представить единственным образом в виде суммы различных меньших степеней двойки, и поэтому наша задача решена. Например, 30 000 = 214 + 213 + 212 + 210 + 28 + 25 + 24.

Но для реального применения нужен алгоритм построения такого разложения. Далее приведем несколько разных алгоритмов, но в начале рассмотрим самый простой. В сущности, это алгоритм выдачи сдачи клиенту, записанный некогда даже в инструкции для работников торговли, но очень редко ими выполняющийся. А он очень прост - сдачу надо выдавать, начиная с самых больших купюр.

**Азбука Морзе**

Сэмюель Морзе известен, однако, не только изобретением азбуки. Он был и художником-портретистом (его картина «Генерал Лафайет» до сих пор висит в нью-йоркском Сити-Холле), и одним из первых фотографов в Америке (учился делать дагерротипные фотографии у самого Луи Дагерра), и политиком (он балатировался в 1836 году на пост мэра Нью-Йорка), но самое главное его достижение - изобретение телеграфа (а азбука Морзе понадобилась ему для использования телеграфа). Заодно он изобрел устройство, которое называется реле. Именно из реле спустя сто лет после Морзе были построены первые компьютеры.

Начал свои работы в этом направление в 1832 году, запатентовал свое изобретение в 1836 году, но публичная демонстрация телеграфа произошла только 24 мая 1844 года. По телеграфной линии, соединяющей Вашингтон с Балтимором, была успешно передана фраза из Библии.

Точка и тире оказались самыми элементарными символами, которые мог передавать его телеграф. Они соответствовали коротким и длинным импульсам электрического тока, передаваемым по телеграфным проводам. Длина импульса определялась нажатием руки телеграфиста на ключ телеграфа. Прием сигнала осуществляло реле, которое после появления в нем импульса тока включало электромагнит, который либо заставлял стучать молоточек, либо прижимал колесико с красящей лентой к бумажной ленте, на которой отпечатывались либо точка, либо тире в зависимости от длины импульса.

Азбука Морзе сопоставляет каждой букве алфавита последовательность из точек и тире. Естественней всего использовать такие последовательности длины 6, их всего 64 и хватит даже на русский алфавит. Но Морзе понимал, что длину сообщения желательно уменьшить, насколько возможно, поэтому он решил использовать последовательности длины не более 4, их всего 2 + 4 + 8 + 16 = 30. в русском алфавите пришлось не использовать буквы «э» и «ё» и отождествить мягкий и твердый знаки. Кроме того, наиболее часто используемых буквами он предложил давать самые короткие коды, чтобы уменьшить среднюю длину передаваемого сообщения. Эту идею в наше время используют с той же целью в алфавитном кодировании.счиние система морзе код

Алфавитное кодирование, штрих-коды и их использование

Пусть, например, кодирующим алфавитом является двухбуквенный алфавит, например, состоящий из символов 0, 1. Схемой алфавитного кодирования называется отображение каждой буквы кодируемого алфавита в некоторое слово в кодирующем алфавите (называемое элементарным кодом), в рассматриваемом случае - последовательность нулей или единиц. Пользуясь этой схемой, можно закодировать любое слово в кодируемом алфавите, заменяя в нем каждую букву на соответствующий ей элементарный код, и превратить исходное слово в более длинное слово в кодирующем алфавите.

Если вместо двоичных цифр использовать обычный алфавит, но со шрифтами двух типов, то таким методом можно в любом тексте спрятать шифровку, если, конечно, шрифты будут достаточно малоразличимы. Желательно при этом использовать разделимый код. Длина зашифрованного сообщения будет в несколько раз короче, чем длина содержащего его (и одновременно маскирующего его) текста, но если для передачи шифровки использовать книгу, то в ней можно, таким образом, незаметно разместить еще целую книгу. Но эта красивая идея из-за дороговизны ее реализации так и не нашла применения. В наше же время ее нельзя рассматривать как серьезный метод.

Примером реального применения двоичного кодирования в современной технике служат штрих-коды. В супермаркетах на упаковках товаров можно увидеть штрих-код. Для чего он нужен, и как его прочитать?

Нужен он только для автоматического занесения информации в кассовый аппарат. Сам штрих-код состоит из тридцати черных полос переменой толщины, разделенной промежутками тоже переменой толщины. Толщина полос может принимать четыре значения - от самой тонкой до самой толстой. Такую же толщину могут иметь и промежутки. Когда по сканеру проводят штрих-кодом, он воспринимает каждую черную полоску как последовательность единиц длины от одной до четырех и также воспринимает промежутки между полосами, но при этом вместо единиц сканер видит нули. Полностью весь штрих-код сканер воспринимает как последовательность из 95 цифр 0 или 1 (их давно уже принято называть битами). Что же содержит этот код? Он кодирует 13-разрядное десятичное число, совершенно открыто написанное под самим штрих-кодом. Какую же информацию содержит это 13-значное число? Этот вопрос к математике никакого отношения не имеет. Первые две цифры задают страну - производителя товара. Следующие пять цифр - это код производитель, а следующие пять цифр - код самого продукта в принятой этим производителем кодировке. Последняя цифра - это код проверки.

Человечество пользуется двоичной системой ежедневно, так как вся информация, обрабатываемая современными ЭВМ, хранится в них в двоичном виде. Каким же образом осуществляется это хранение? Каждый регистр арифметического устройства ЭВМ, каждая ячейка памяти представляет собой физическую систему, состоящую из некоторого числа однородных элементов. Каждый такой элемент способен находиться в нескольких состояниях и служит для изображения одного из разрядов числа. Именно поэтому каждый элемент ячейки называют разрядом. Нумерацию разрядов в ячейке принято вести справа налево, самый левый разряд имеет порядковый номер 0. Если при записи чисел в ЭВМ мы хотим использовать обычную десятичную систему счисления, то мы должны получать 10 устойчивых состояний для каждого разряда, как на счетах при помощи костяшек. Такие машины существуют. Однако конструкция элементов такой машины чрезвычайно сложна. Наиболее надежным и дешевым является устройство, каждый разряд которого может принимать два состояния: намагничено — не намагничено, высокое напряжение — низкое напряжение и т. д. В современной электронике развитие аппаратной базы ЭВМ идет именно в этом направлении. Следовательно, использование двоичной системы счисления в качестве внутренней системы представления информации вызвано конструктивными особенностями элементов вычислительных машин.

В данный момент для кодировки информации в компьютере используется двоичная система счисления. Каждый символ в компьютере представляется в виде последовательности единиц и нулей, любая такая последовательность состоит из восьми знаков. Знакомства в таких последовательностях называется битом, а восемь битов это байт.

Для перевода значений отдельных байтов в понятные человеку знаки (буквы и цифры) компьютер использует специальные «кодовые таблицы», в которых каждому знаку сопоставлен байт с определенным значением.

Впрочем, измерять компьютерную информацию байтами весьма неудобно из-за объема. Вот почему на практике в компьютерном мире оперируют такими величинами: • килобайт (кб) — 2 в степени 10 байт — 1024 байт;

• мегабайт (Мб) — 2 в степени 20 байт — 1 048 576 байт —1024 кб;

• гигабайт (Гб) — 2 в степени 30 байт — 1 073 741 824 байт -

1 048 576 кб-1024 Мб;

• терабайт (Тб) — 2 в степени 40 байт — 1 099 511 627 776 байт —

1 073 741 824 кб - 1 048 576 Мб - 1024 Гб;

• петабайт (Пб) - 2 в степени 50 байт - 1125 899 906 842 624 байт -

1 099 511 627 776 кб - 1073 741 824 Мб - 1 048 576 Гб - 1024 Тб

Биты используются в компьютерной терминологии значительно реже, — например, в показателях скорости передачи данных:

• килобит (кбит) — 2 в степени 10 бит —'1024 бит — 128 байт;

• мегабит (Мбит) — 2 в степени 20 бит — 1 048 576 бит — 1024 кбит-128 кб;

• гигабит (Гбит) — 2 в степени 30 бит — 1 073 741 824 бит -

1 048 576 кбит - 1024 Мбит - 128 Мб.

Как было уже сказано, все числовые данные хранятся в машине в двоичном виде, то есть в виде последовательности нулей и единиц, однако формы хранения целых и вещественных чисел различны.

Целые числа хранятся в форме с фиксированной запятой, вещественные числа хранятся в форме с плавающей запятой. Существует большой класс задач, в которых не используются вещественные числа. Например, задачи экономического характера, при решении которых данными служат количество деталей, акций, сотрудников и так далее, работают только с целыми числами. Текстовая, графическая и звуковая информация, как это будет показано ниже, также кодируются в компьютере с помощью целых чисел. Для повышения скорости выполнения таких задач и используется представление целых чисел в форме с фиксированной запятой.

Более того, в современных персональных компьютерах процессоры выполняют операции только над целыми числами в форме с фиксированной запятой.

**Перевод различных видов систем счисления.**

Перевод из произвольной позиционной системы счисления в десятичную систему.

Для перевода из любой позиционной системы счисления в десятичную систему используется следующий алгоритм:

Пронумеруем цифры в изначальной записи числа справа налево, начиная с нуля (номера соответствуют степени основания в многочлене )

Умножим каждое число на соответствующую степень основания.

Складываем получившиеся произведения.

Приведем пример:

11012 =1\*23 + 1\*22 + 0\*21+ 1\*20= 8+4+0+1=1310

3 2 1 0

1204205= 1\*55+2\*54+0\*53+4\*52+2\*51+0\*50= 3125+1250+0+100+10+0=448510

Перевод из десятичной системы в произвольную позиционную систему счисления

Для перевода из десятичной системы счисления в любую позиционную необходимо придерживаться следующего алгоритма:

1. Делим исходное число на основание нацело в десятичной системы счисления и записываем в качестве нового значения десятичного целую часть результата от деления.

2. Остаток от деления (он должен быть не больше основания данной системы) записываем начиная с последнего.

Приведем пример:

4410 переведём в двоичную систему

44 делим на 2. частное 22, остаток 0

22 делим на 2. частное 11, остаток 0

11 делим на 2. частное 5, остаток 1

5 делим на 2. частное 2, остаток 1

2 делим на 2. частное 1, остаток 0

1 делим на 2. частное 0, остаток 1

Частное равно нулю, деление закончено. Теперь записав все остатки, справа налево получим число 1011002

Перевод в машинной группе.

Для этого типа операций существует упрощенный алгоритм.

Для восьмеричной — разбиваем число на триады, для шестнадцатеричной — разбиваем на тетрарды, преобразуем триады по таблице

Пример: преобразуем 1011002 восьмеричная — 101 100 → 548 шестнадцатеричная — 0010 1100 → 2C16

Обратный перевод из восьмеричной и шестнадцатеричной в двоичную осуществляется за счет замены цифр соответствующими триадами и тетрардами.

548 → 101 1002

2C16 → 0010 11002

Дробное счисление в других системах счисления

До этого в рассмотренных примерах показателем степени основания системы счисления являлось натуральное число, но ничто не мешает перевести показатель степени в диапазон целых чисел, т. е. расширить его в отрицательную полуплоскость. При этом формула, данная в определении будет, также верна.

Рассмотрим пример: число 103,625 можно представить как

Таким образом, из примера видно, что не только целое, но и дробное число можно представить как комбинацию из цифр системы счисления.

Перевод из произвольной системы счисления в десятичную систему.

Рассмотрим пример перевода двоичного числа 1100,0112 в десятичное систему. Целая часть этого числа равна 12 (см. выше), а вот перевод дробной части рассмотрим подробнее:

Итак, число 1100,0112 = 12,37510.

Точно также осуществляется перевод из любой системы счисления, только вместо «2» ставится основание системы.

Для удобства перевода, целую и дробную части числа почти всегда переводят по отдельности, а результат потом суммируют.

Перевод из двоичной системы в 8- и 16-ричную

Перевод дробной части из двоичной системы счисления в системы счисления с основаниями 8 и 16 осуществляется точно так же, как и для целых частей числа, за тем лишь исключением, что разбивка на триады и тетрады идёт вправо от десятичной запятой, недостающие разряды дополняются нулями справа. Например, рассмотренное выше число 1100,0112 будет выглядеть как 14,38 или C,616.

Перевод из десятичной системы в произвольную систему

Для перевода дробной части числа в другие системы счисления нужно обратить целую часть в нуль и начать умножение получившегося числа на основании той системы, в которую нужно перевести. Если в результате умножения будут снова появляться целые части, их нужно повторно обращать в нуль, предварительно запомнив (записав) значение получившейся целой части. Операция заканчивается, когда дробная часть полностью обратится в нуль. Ниже приводится пример перевода числа 103,62510 в двоичную систему счисления.

Переводим целую часть по правилам, описанным выше, получаем 10310 = 11001112.

0,625 умножаем на 2. Дробная часть 0,250. Целая часть 1.

0,250 умножаем на 2. Дробная часть 0,500. Целая часть 0.

0,500 умножаем на 2. Дробная часть 0,000. Целая часть 1.

Итак, сверху вниз получаем число 1012

103,62510 = 1100111,1012

Точно также осуществляется перевод в системы счисления с любым основанием.

Сразу нужно отметить, что этот пример специально подобран, в общем случае очень редко удаётся завершить перевод дробной части числа из десятичной системы в другие системы счисления, а потому, в подавляющем большинстве случаев, перевод можно осуществить с какой либо долей погрешности. Чем больше знаков после запятой — тем точнее приближение результата перевода к истине. В этих словах легко убедиться, если попытаться, например, перевести в двоичный код число 0,626.

112 и восьмеричное число 554768 на 58:

-110110 11 55476 5

11 10010 - 5 11077

- 0011 -05

00 - 047

Так же для работы используются соответствующие таблицы умножения.

**Использование систем счисления в компьютерной технике**

**и информационных технологиях.**

Столь привычная для нас десятичная система оказалась неудобной для ЭВМ. Если

в механических вычислительных устройствах, использующих десятичную систему, достаточно просто применить элемент с множеством состояний (колесо с девятью зубьями), то в электронных машинах надо было бы иметь 10 различных потенциалов в цепях. Наиболее просто реализуется элементы с двумя состояниями - триггеры. Поэтому естественным был переход на двоичную систему, т.е. системы по основанию 2. В этой системе всего две цифры - 0 и 1 . Каждая цифра называется двоичной (от английского binarydigit - двоичная цифра). Сокращение от этого выражения привело к появлению термина бит, ставшего название разряда двоичного числа.

Бит - это минимальная единица измерения информации (0 mini). За битом следует байт, состоящий из восьми бит, затем килобайт (кбайт) - 1024 байта, мегабайт (мбайт) - 1024 кбайта, гигобайт (гбайт) - 1024мбайт.

Двоичное кодирование информации в компьютере

Для обмена информацией с другими людьми человек использует естественные языки (русский, английский, китайский и т.д.), то есть информация представляется с помощью естественных языков. В основе языка лежит алфавит, то есть набор символов (знаков, которые человек различает по их начертанию. Последовательности символов алфавита в соответствии с правилами грамматики образуют основные объекты языка -слова. Правила, согласно которым образуются предложения из слов данного языка называются синтаксисом. Необходимо от метить, что в естественных языках грамматика и синтаксис языка формируются с помощью большого количество правил, из которых существуют исключения, так как такие правила складывались исторически.

Наряду с естественными языками были разработаны формальные языки (системы счисления, язык алгебры, языки программирования и др.). Основное отличие формальных языков от естественных состоит в наличии жестких правил грамматики и синтаксиса. Эти языки были разработаны людьми, для упрощения каких-либо действий. Как, например, системы счисления были придуманы для упрощения подсчетов чего-либо.

Создатели первых компьютеров столкнулись с проблемой представления и обработки информации. Так как компьютер это всего лишь машина у которой нет ни интеллекта, ни логики, и мыслить она не способна, разработчикам пришлось найти такой способ представления информации, который был бы максимально прост для восприятия компьютером. Столь привычная для нас десятичная система оказалась неудобной для ЭВМ. Если в механических вычислительных устройствах, использующих десятичную систему, достаточно просто применить элемент с множеством состояний (колесо с девятью зубьями), то в электронных машинах надо было бы иметь 10 различных потенциалов в цепях. Наиболее просто реализуется элементы с двумя состояниями - триггеры. Поэтому естественным был переход на двоичную систему, т.е. системы по основанию 2. В этой системе всего две цифры - 0 и 1 . Каждая цифра называется двоичной (от английского binarydigit - двоичная цифра). Сокращение от этого выражения привело к появлению термина бит, ставшего названием разряда двоичного числа.

Бит - это минимальная единица измерения информации (0 или1). За битом следует байт, состоящий из восьми бит, затем килобайт (кбайт) - 1024 байта, мегабайт (мбайт) - 1024 кбайта, гигобайт (гбайт) - 1024мбайт.

Таким образом, в компьютере для представления информации используется двоичное кодирование, так как удалось создать надежные работающие технические устройства, которые могут со стопроцентной надежностью сохранять и распознавать не более двух различных состояний (цифр). Все виды информации в компьютере кодируются на машинном языке, в виде логических последовательностей нулей и единиц.

Например целое неотрицательное число А2=Т 111100002 будет храниться в ячейке следующим образом:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |

Значит, мы можем записать все числа от 0 до 255 в двоичной системе счисления в 1 ячейке памяти.

Представление чисел в компьютере

Целые числа в компьютере хранятся в ячейках памяти, в этом случае каждому разряду ячейки памяти соответствует всегда один и тот же разряд числа.

Для хранения целых неотрицательных чисел отводится одна ячейка памяти состоящая из восьми бит.

Например, число 1910 будет выглядеть:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | о | 0 | 1 | 0 | 0 | 1 | 1 |

Для хранения целых чисел со знаком (отрицательных) отводиться две ячейки памяти (16 битов), причем старший (левый) разряд отводиться под знак числа (если число положительное, то в знаковый разряд записывается 0, если отрицательное - 1).

Например, число -9810 будет выглядеть:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 |

Представление целых чисел в компьютере в формате «знак - величина» называется прямым кодом числа. Например, число 200210=111110100102 будет представлено в шестнадцатиразрядном представлении следует следующим образом:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |

Для представления отрицательных чисел используется дополнительный код. Дополнительный код позволяет заменить арифметическую операцию

вычитания операцией сложения, что существенно упрощает работу процессора и увеличивает его быстродействие. Дополнительный код представляет собой дополнение модуля отрицательного числа А до О,

2n-|А|+|А| =0

поскольку в компьютерной 2n=0.

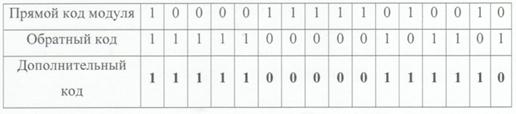
Алгоритм построения такого кода довольно прост:

1. Записать модуль числа в прямом коде.

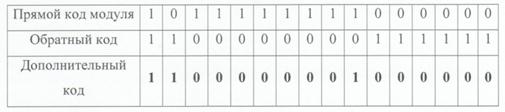
2. Получить обратный код числа (то есть заменить все нули на единицы, а все единицы на нули).

3. К полученному результату прибавить единицу.

Запишем дополнительный код отрицательного числа -2002 для 16-разрядного представления:



Запишем дополнительный код отрицательного числа -16320 для 16-разрядного представления:



Способы построения двоичных кодов

Начиная с конца 60-х годов, компьютеры все больше использовать для обработки текстовой информации и в настоящее время большая часть компьютеров в мире занято именно обработкой текстовой информации.

Традиционно для кодирования одного символа используется количество информации равное 1 байту, то есть 8 бит. Если рассматривать символы как возможные события, то получаем, что количество различных символов, которые можно закодировать, будет равно 256. Такое количество символов вполне достаточно для представления текстовой информации, включая прописные и строчные буквы русского и латинского алфавитов, а так же цифры, знаки препинания и математических операций, графические символы и так далее.

Но способов построения таких кодов очень много, рассмотрим некоторые из них:

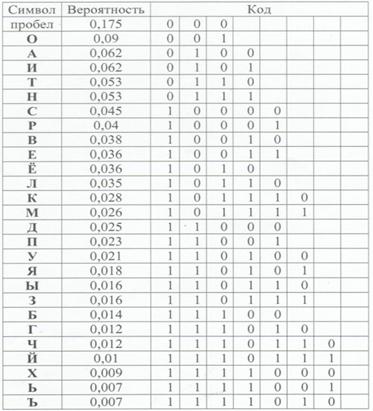
Алфавитное неравномерное двоичное кодирование

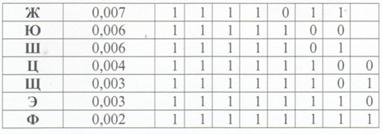
При алфавитном способе двоичного кодирования символы некоторого первичного алфавита (например, русского) кодируются комбинациями символов двоичного алфавита (т.е. 0 и 1), причем, длина кодов и, соответственно, длительность передачи отдельного кода, могут различаться. Оптимизировать кодирование можно за счет суммарной длительности сообщения.

Суммарная длительность сообщения будет меньше, если применить следующий подход: чем буква первичного алфавита, встречается чаще, то присваиваем ей более короткой по длине код. Следовательно, коды букв, вероятность появления которых в сообщении выше, следует строить из возможно меньшего числа элементарных сигналов.

Возможны различные варианты двоичного кодирования, однако, не все они будут пригодны для практического использования - важно, чтобы закодированное сообщение могло быть однозначно декодировано*,* т.е. чтобы в последовательности *0* и 1, которая представляет собой многобуквенное кодированное сообщение, всегда можно было бы различить обозначения отдельных букв.

Рассмотрим пример построения двоичного кода для символов русского алфавита:





Неравномерный код с разделителями

Для того что бы было проще декодировать сообщения был придуман код с разделителями.

Проще всего достичь однозначного декодирования, если коды будут разграничены разделителем *-* некоторой постоянной комбинацией двоичных знаков. Условимся, что разделителем отдельных кодов букв будет последовательность 00(признак конца знака), а разделителем слов - 000(признак конца слова - пробел). Довольно очевидными оказываются следующие правила построения кодов:

код признака конца знака может быть включен в код буквы, поскольку не существует отдельно (т.е. кода всех букв будут заканчиваться 00);

коды букв не должны содержать двух и более нулей подряд в

середине (иначе они будут восприниматься как конец знака);

код буквы (кроме пробела) всегда должен начинаться с 1;

разделителю слов (000)всегда предшествует признак конца знака;

при этом реализуется последовательность 00000(т.е. если в конце кода встречается комбинация ...000или ...0000,они не воспринимаются как разделитель слов); следовательно, коды букв могут оканчиваться на 0 или 00 (до признака конца знака).

Длительность передачи каждого отдельного кода 4 очевидно, может быть найдена: ti = ki • τ, где ki- количество элементарных сигналов (бит) в коде символа *L.*

Алфавитное неравномерное двоичное кодирование. Префиксный код

Рассмотрев один из вариантов двоичного неравномерного кодирования, попробуем найти ответы на следующие вопросы: возможно ли такое кодирование без использования разделителя знаков? Существует ли наиболее оптимальный способ неравномерного двоичного кодирования?

Суть первой проблемы состоит в нахождении такого варианта кодирования сообщения, при котором последующее выделение из него каждого отдельного знака (т.е. декодирование) оказывается однозначным без специальных указателей разделения знаков. Наиболее простыми и употребимыми кодами такого типа являются так называемые префиксные коды*,* которые удовлетворяют следующему условию (условию Фано)*:*

Неравномерный код может быть однозначно декодирован, если никакой из кодов не совпадает с началом какого-либо иного более длинного кода.

Например, если имеется код  ПО, то уже не могут использоваться коды 1, 11, 1101, 110101 и пр. Если условие Фано выполняется, то при прочтении (расшифровке) закодированного сообщения путем сопоставления со списком кодов всегда можно точно указать, где заканчивается один код и начинается другой.

Пример: Пусть имеется следующая таблица префиксных кодов:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| а | л | м | р | у | ы |
| 10 | 010 | 00 | 11 | 0110 | 0111 |

Требуется декодировать сообщение: 00100010000111010101110000110. Декодирование производится циклическим повторением следующих действий:

1. отрезать от текущего сообщения крайний левый символ, присоединить к рабочему кодовому слову;

2. сравнить рабочее кодовое слово с кодовой таблицей; если совпадения нет, перейти к (1);

3. декодировать рабочее кодовое слово, очистить его;

4. проверить, имеются ли еще знаки в сообщении; если «да», перейти к (1).

Применение данного алгоритма дает:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Шаг | Рабочее слово | Текущее сообщение | Распознанный знак | Декодированное сообщение |
| 0 | пусто | 00100010000111010101110000110 | - | - |
| 1 | 0 | 0100010000111010101110000110 | нет | - |
| 2 | 00 | 100010000111010101110000110 | м | М |
| 3 | 1 | 00010000111010101110000110 | нет | М |
| 4 | 10 | 0010000111010101110000110 | а | МА |
| 5 | 0 | 010000111010101110000110 | нет | МА |
| 6 | 00 | 10000111010101110000110 | м | Мам |
| • • • |  |  |  |  |

Доведя процедуру до конца, получим сообщение: «мама мыла раму».

Код Хаффмана

Способ оптимального префиксного двоичного кодирования был предложен Д.Хаффманом. Построение кодов Хаффмана мы рассмотрим на следующем примере: пусть имеется первичный алфавит А, состоящий из шести знаков a1…а6с вероятностями появления в сообщении, соответственно, 0,3; 0,2; 0,2; 0,15; 0,1; 0,05. Создадим новый вспомогательный алфавит Аb объединив два знака с наименьшими вероятностями (а5 и а6) и заменив их одним знаком (например, а(1)); вероятность нового знака будет равна сумме вероятностей тех, что в него вошли, т.е. 0,15; остальные знаки исходного алфавита включим в новый без изменений; общее число знаков в новом алфавите, очевидно, будет на 1 меньше, чем в исходном. Аналогичным образом продолжим создавать новые алфавиты, пока в последнем не останется два знака; ясно, что число таких

шагов будет равно N - 2, где N - число знаков исходного алфавита (в нашем случае N = 6, следовательно, необходимо построить 4 вспомогательных алфавита). В промежуточных алфавитах каждый раз будем переупорядочивать знаки по убыванию вероятностей. Всю процедуру построения представим в виде таблицы:



Теперь в обратном направлении поведем процедуру кодирования. Двум знакам последнего алфавита присвоим коды 0 и 1 (которому какой - роли не играет; условимся, что верхний знак будет иметь код 0, а нижний - 1). В нашем примере знак а1(4) алфавита А(4), имеющий вероятность 0,6 , получит код 0, а а2(4) с вероятностью 0,4 - код 1. В алфавите A(3) знак а1(3) с вероятностью 0,4 сохранит свой код (1); коды знаков a2(3) и a3(3), объединенных знаком a1(4) с вероятностью 0,6 , будут уже двузначным: их первой цифрой станет код связанного с ними знака (т.е. 0), а вторая цифра -как условились - у верхнего 0, у нижнего - 1; таким образом, а2(3) будет иметь код 00,  a a3(3) - код 01.  Полностью  процедура кодирования

представлена в следующей таблице:



Из самой процедуры построения кодов легко видеть, что они удовлетворяют условию Фано и, следовательно, не требуют разделителя. Средняя длина кода при этом оказывается: К(2) = 0,3-2+0,2-2+0,2-2+0,15-3+0,1-4+0,05-4 = 2,4

Можно заключить, что существует метод построения оптимального неравномерного алфавитного кода. Метод Хаффмана и его модификация - метод адаптивного кодирования (динамическое кодирование Хаффмана) - нашли применение в программах-архиваторах, программах резервного копирования файлов и дисков, в системах сжатия информации в модемах и факсах.

Равномерное алфавитное двоичное кодирование. Байтовый код

В этом случае двоичный код первичного алфавита строится цепочками равной длины, т.е. со всеми знаками связано одинаковое количество информации равное 10. Передавать признак конца знака не требуется, поэтому для определения длины кодовой цепочки можно воспользоваться формулой: К(2) > log2N. Приемное устройство просто отсчитывает оговоренное заранее количество элементарных сигналов и интерпретирует цепочку (устанавливает, какому знаку она соответствует). Правда, при этом недопустимы сбои, например, пропуск (непрочтение) одного элементарного сигнала приведет к сдвигу всей кодовой последовательности и неправильной ее интерпретации; решается проблема путем синхронизации передачи или иными способами. С другой стороны, применение равномерного кода оказывается одним из средств контроля правильности передачи, поскольку факт поступления лишнего элементарного сигнала или, наоборот, поступление неполного кода сразу интерпретируется как ошибка.

**ЗАКЛЮЧЕНИЕ**

В какой системе счисления лучше записывать числа - это вопрос удобства и традиций. С технической точки зрения, в ЭВМ удобно использовать двоичную систему, так как в ней для записи числа используется всего две цифры 0 и 1, которыми можно представить двумя легко различимыми состояниями «нет сигнала» и «есть сигнал».

Изучая источники по теме «Системы счисления» мы получили возможность провести исторический анализ, исследовать различные формы записи чисел, систематизировать материал и выявить различные спектры применения.

Различные системы счисления окружают нас повсюду. Сами того не замечая мы ежедневно пользуемся не только десятичной системой счисления, а так же двенадцатеричной, когда хотим узнать время или покупаем в магазине пуговицы.

Сейчас системы счисления очень распространены в электронно-вычислительной технике, многие коды и шифры созданы на их основе.

В ходе проведения исследования:исследовали историю и развитие систем счисления,исследовали практический материал, рассмотрели область применения и выявили актуальность темы.

Нами решены задачи:

арифметические действия в различных системах счисления,

— перевод из одной системы счисления в другую. Нашу работу можно использовать на уроках информатики в качестве обучающего материала, так как мы, выполняя деление понятия, старались соблюдать принципы.

**Блиц-опрос.**

1. Цифра –это…?
2. Виды систем счисления?
3. Дать определение непозиционным СС, привести пример;
4. Дать определение позиционным СС, привести пример;
5. Что мы называем позицией системы счисления?
6. Что такое бит?
7. Сколько цифр содержит десятичная , восьмеричная, шестнадцатеричная и двоичная СС?

**СПИСОК ЛИТЕРАТУРЫ**

1. Алгебра и теория чисел: Учеб.пособие для студентов-заочников II курса физ.-мат. фак. пед. ин-тов (Н.А.Казачёк и др.) / Под ред. Н.Я. Виленкина - 2-е изд. М.: Просвещение, 1984. - 192 с.
2. Бендукидзе А.Д. О системах счисления // Квант - 1975 - №8 - с 59-61.
3. Берман Г.Н. Число и наука о нем. Общедоступные очерки по арифметики натуральных чисел. Изд. 3-е. М.: Физматгиз, 1960. - 164с.
4. Вайман А.А. Шумеро-вавилонская математика. III - I тысячелетия до н.э. М.: Изд. вост. лит., 1961. - 278с.
5. Выгодский М.Я. Арифметика и алгебра в древнем мире. Изд. 2-е, испр. идоп. М.: Наука, 1967. - 367 с.
6. Глейзер Г.И. История арифметике в школе: IV - VI кл. Пособие для учителей. - М.: Просвещение, 1981. - 239 с.
7. Гутер Р.С. Вычислительные машины и системы счисления // Квант-1971 -№2.
8. Депман И.Я. История арифметики, пособие для учителей. М.: Учпедгиз, 1959.-423с.
9. Депман И.Я., Виленкин Н.Я. За страницами учебника математики: Пособие для учащихся 5-6 кл. сред.шк. М.: Просвещение, 1989. -287с.
10. Детская энциклопедия: [В 10-ти т.] Для среднего и старшего возраста. Гл.ред. Маркушевич А.И. Т.2. - Мир небесных тел; Числа и фигуры. -М.: Педагогика, 1972. - 480 с.