# ДЗ на 13.09.2024

1. Сколько базисных знаков в римской СС можно писать подряд?

**Ответ**: 3 базисных знака.

1. Определить порядок действий в римской СС

**Ответ**:

* 1. Чтение осуществляется слева направо, как и запись.
  2. Когда из двух рядом стоящих чисел записывается меньшее слева, то итог отнимается (IX = 10 - 1 = 9).
  3. Верно и противоположное с записью справа и сложением (DI = 500 + 1 = 501).
  4. Одну и ту же цифру не используют более трёх раз подряд (например, не IIII, a IV).
  5. Вычитание происходит только со степенями десятки.
  6. Вычитание происходит только один раз при записи разряда.
  7. При переводе из наших привычных чисел уменьшаемое число не должно превышать более чем в десять раз, то есть число девять мы можем представить как IX, а вот девяносто девять как IC уже не получится

1. Будут ли отличаться записи знаков, если сверху поставить черточку?

**в**

1. Элементная база современных ЭВМ?

**Ответ**:

# Перевод из десятичной в любую другую систему счисления с основанием P

## Перевод целой части из десятичной СС. Алгоритм

1. Целая часть числа делится на основание системы счисления P до получения целого частного. Полученный при этом остаток (в том числе и 0) будет младшим разрядом числа в новой СС.
2. Полученное частное снов делим на P до тех пор, пока частное не станет меньше P.
3. Полученный при этом остатки будут разрядами числа в новой СС.
4. Число в новой СС получается путем выписывания в обратном порядке последнего частного и полученных остатков.

## Перевод из десятичной СС в любую другую дробной части. Алгоритм

1. Дробная часть последовательной умножается на основание P новой СС.
2. Полученные при этом целые части чисел будут разрядами числа в новой СС.
3. Умножение прекращается по одной из двух причин: 1) Дробная часть стала равна нулю; 2) Достигнута заданная точность вычисления (4 знака после запятой)

ДЗ: перевести число 47.37 в три другие СС. 470.5146 из восьмеричной в три других (в десятичную по Горнеру, мантису двумя способами)

239.87510 => 11101111.1112 => 357.78 => EF.E16.

## Смешанные системы счисления

Числа, представленные в системах счисления с основаниями P и Q связанные формулой P = QX где P – основание первой системы счисления, Q – основание второй системы счисления, а K – количество разрядов, которое связывает P и Q. В таких СС (связанных) числа можно переводить по таблице.

|  |  |  |  |
| --- | --- | --- | --- |
| **Десятичная СС** | **Восьмеричная СС** | **Шестнадцатеричная СС** |  |
| 0 | 000 | 0000 |  |
| 1 | 001 | 0001 |  |
| 2 | 010 | 0010 |  |
| 3 | 011 | 0011 |  |
| 4 | 100 | 0100 |  |
| 5 | 101 | 0101 |  |
| 6 | 110 | 0110 |  |
| 7 | 111 | 0111 |  |
| 8 |  | 1000 |  |
| 9 |  | 1001 |  |
| 10 |  | 1010 | A |
| 11 |  | 1011 | B |
| 12 |  | 1100 | C |
| 13 |  | 1101 | D |
| 14 |  | 1110 | E |
| 15 |  | 1111 | F |

46.5 10 => X 2,8,16  = 56.4 8 => 101110.100 2 => 2E.8 16

175.25 8 => 001111101.010101 2 => 7D.54 16 => 125.3281 10

ДЗ: число E2E4.258 из 16 в три других (мантиса 2 горнера)

# Числовая система компьютера

ЧСК – связанна с системой кодирования информации. Кодирование – переход от одного представления информации (Удобного для восприятия человеком) к представлению, удобному для хранения, передачи и обработки компьютером.

## Цели кодирования:

1. Удобство технической реализации.
2. Удобство восприятия стороной создания кода.
3. Высокая скорость передачи и обработки.
4. Экономичность систем кодирования (отсутствие синонимов).
5. Надежность – защита от случайных искажений.
6. Сохранность информации – защита от нежелательного доступа.

## Противоречия в системах кодирования:

1. Цель экономичности уменьшает удобство системы и надежность.
2. Короткое сообщение (экономичность) может увеличить скорость обработки, но уменьшить объем обрабатываемой информации.
3. Цель защиты информации может не только объем хранимой информации, но и замедлить работу с ней

Применяется на разных этапах обработки разные цели. Поэтому информация неоднократно при переходе с этапа на этап перекодировывается.

## Представления чисел в компьютере

Количество информации, которое может помещаться в один элемент памяти (0 или 1) называется бит. Оно очень мало и не несет смысловой нагрузки. 1 байт = 8 бит.

Машинное слово – элементы памяти – байты (порядковый номер байта называется его адресом) – соединяются в одну ячейку, которую аппаратная часть компьютера воспринимает как единое целое, например 32, 64 бита. Таким образом, объем информации, который представлен в данном компьютере ограничен емкостью памяти, следовательно числовая информация может представляться лишь с определенной степенью точности, это зависит от длинны машинного слова.

Адрес машинного слова равен адресу младшего байта этого слова. Нумерация байтов памяти начинается с 0.

Дано: объем ОЗУ = 2кб. Какой номер имеет адрес последнего байта, если нумерация байтов идет с 0.

Решение: Переводим емкость в байты: 2кб = 2\*2 10 = 2048 байтов. Так как нумерация идет с 0, то номер последнего байта = 2048 -1.

Дано: оперативная память компьютера составляет 163840 машинных слов. В объеме это составляет 0,625 Мб. Какова длинна машинного слова?

Решение:

0,625 Мб = 640 Кб = 655360 байт = 5 242 880 бит

1 слово = 5242880/163840 = 32 бита.

Дано: объем ОЗУ 1 Мб. Адрес последнего МС равен 1048574. Чему равен размер машинного слова?

Решение:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 байт | |  |  |  |  | | --- | --- | --- | --- | | 1бит | 1бит |  |  | |  |  |  |  | | 2 байт | 3 |  |  |  |  |

Ответ: МС = 2 байта.

## Форматы представления числовой информации

### Числа с фиксированной точкой

Применяется только для целых чисел.

Целые числа – диапазон целых чисел зависит от размера машинного слова данного компьютера.

Дано: Используется k-разрядное машинное слово (k-битовое машинное слово). Тогда определить диапазон хранимых целых чисел в двух вариантах: а) если хранятся только положительные числа б) хранятся и положительные и отрицательные числа в равном количестве.

Решение:

K=16 бит. Тогда всего в машинном слове может хранится 2 16 = 65536 различных чисел. => а) 65535 чисел б) 65536 / 2 = 32768 различных значений => -32768 <=X<= 32767.

#### Алгоритм представления целого положительного числа N в компьютере

1. Число N переводится в двоичную систему счисления.
2. Если машинное слово имеет k разрядов, то полученное на шаге 1 число вписывается в машинное слово начиная с последнего разряда.
3. Оставшиеся разряды машинного слова дополняются слева незначащими нулями
4. Если применяется упакованная форма хранения, то полученное значение в машинном слове переводится в 16 систему счисления.

Дано: целое положительное число N = 1607 10 => X2 = 1607/2 = 11001000111

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

Упакованная форма: 0647 16

ДЗ: Перевести из 10 СС во внутреннее представление +263, +195

Задание: представить в машинном слове +234, +98. Найти упакованную форму

Дано: 2 байт машинное слово ощущение

Решение:

1)

+234 10 = 1110 1010 2 = EA 16

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

Упакованная форма: 00EA

2)

+98 = 01100010 2 = 62 16

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

Упакованная форма: 0062

#### Алгоритм внутреннего компьютерного представления отрицательного числа -N

1. Получаем внутреннее компьютерное представление числа в двоичной СС.
2. Получаем обратный код этого числа, заменяя 0 на 1, а 1 на 0.
3. К полученному числу добавляем 1 и получаем дополнительный код.
4. Получаем упакованную форму в 16 СС.

-1607 10 = 11001000111 2 =

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

=

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

=

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

Упакованная форма: F9B9 16

-234, -98

А)

-234 =

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

=

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

=

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

Упакованная форма: FF16

Б)

-98 =

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

=

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

=

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

Упакованная форма: FF9E

+-222172, 4 байт

+222172 10 = 0011 0110 0011 1101 1100 2 =

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

Упакованная форма: 000363DC

-222172 10 =

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

=

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

Упакованная форма: FFFC9C24 16

### Внутреннее компьютерное представление действительных чисел

Любое действительное число в компьютере может быть представлено в виде формулы:

R=+-(m\*n +-p )

Где R – действительное число, m – мантисса числа, n – основание СС в которой считается число, p – порядок числа.

R = +25.384 = 253.84\*10 -1 = 0.0025384 \* 10 4 = 0.25384 \* 10 2

Что хранить:

1. Знак числа
2. Все цифры числа по формуле 1 (0.цифры\_мантиссы\*10 p, , первая цифра != 0 )
3. Знак порядка p
4. Значение порядка

Что не хранить:

1. Основание СС
2. **0.**цифры

#### Алгоритм представления действительного числа

Дано: машинное слово длинной 4 байта.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | | Хранит знак числа. 0 если +, 1 если - |  |  |  |  |  |  |  | | Машинный порядок числа.  Min = 0000000 = 0,  Max = 1111111 = 127,  Суммарно 128 значений => 64 – отрицательные порядки (от -64 до -1), 64 – положительные порядки (от 0 до 63) => -64 до 63 | | | | | | |   1 байт.  Адрес - 0 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |   2 байт  Адрес - 8 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |   3 байт  Адрес - 16 | |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | |  |  |  |  |  |  |  |  |   4 байт  Адрес - 24 |
| Тут знак и порядок | Тут хранится мантисса. | | |

Математический и машинные порядки учитывают то, что машинные порядки имеют только положительные значения (от 0 до 127). Следовательно, m и p сдвинуты относительно друг друга, так что минимальному математическому порядку (-64(зависит от разрядности)) соответствует 0 машинного порядка. Следовательно, в алгоритм связи машинных и математических порядков должна быть заложена формула

M 10 = p 10 + 64 10

=> Избавились от хранения знака порядка.

M 2 = p 2 + 1000000 2

##### Шаги алгоритма

1. Модуль заданного числа переводится в двоичную СС с 24 (3 бита) значащими цифрами.
2. Число в двоичной СС представляется в нормализованной форме по формуле 1
3. Находится машинный порядок M в двоичной СС по формуле 3
4. Вписывается знак числа в младший бит машинного слова
5. Учитывая знак числа, оно вписывается в формат (например 4 байта) машинного слова
6. Представляется упакованная форма

Дано число 250.1875 10 . Перевести в 2 СС

1. 1111 1010. 0011 2 = 1111 1010 . 0011 0000 0000 0000
2. Нормализованная форма: 0.1111 1010 0011 0000 0000 0000 2 \* 10 2 ^ 1000 2
3. Вычислить порядок по формуле 3. M 2= 1000 2 + 100 0000 2 = 1001000 2
4. Записать в байт: 0100 1000 1111 1010 0011 0000 0000 0000
5. Получить упакованную форму: 48FA3000 16

Представить число 37,291 10 в 2 СС

1. 37,291 = 10 0101.0100 1010 0111 1110 11 2
2. Нормализованная форма: 0.1001 0101 0010 1001 1111 1011 2 \* 10 2 ^ 110 2
3. M 2 = 110 2 + 100 0000 = 100 0110 2
4. Записать в байт: 0100 0110 1001 0101 0010 1001 1111 1011 2
5. Получить упакованную форму: 469529FB 16

-370,25 в 2СС

1. 370,25 = 1 0111 0010,01 2
2. Нормализованная форма: 0,1011 1001 0010 2 \* 10 2 ^ 1001 2
3. M 2 = 1001 2 + 0100 0000 2 = 0100 1001 2
4. Записать в байт: 1100 1001 | 1011 1001 | 0010 0000 | 0000 0000
5. Получить упакованную форму: C9B92000 16

Дано: C981 1000

1. Перейти к двоичному представлению числа, вписанного в машинное слово размером 4 байта 1100 1001 1000 0001 0001 0000 0000 0000
2. Найдем математический порядок числа из формулы M = P + 100 0000 2 => P = M – 100 0000 = 1001
3. Запишем число в нормализованной форме: -0,1000 0001 0001 0000 0000 0000 \* 10 ^ 1001
4. Представим число с выделением целой части: -1 0000 0010,001 = 258,125
5. R = -258,125

## Представление символьной информации

Представление идет через американский стандартный код обмена информации ASCII. Он генерируется внешними устройствами (например принтером) и используется для обмена данными между ними и оперативной памятью.

Форматом представления символьной информации являются:

1. Логические коды, которые могут размещаться: в отдельных байтах; в отдельных машинных словах. Логическими кодами представляются: символы; числа без знака и битовые величины. Обычно 1 символ в ASCII занимает 1 байт памяти
2. Числа с фиксированной точкой без знака. При однобайтовом представлении символов в ASCII всего может быть 256 символов, где символы от 0-го до 127-го содержат: символы латинского алфавита; цифры; знаки препинания; скобки и служебные символы. Символы со 128-го до 255-го содержат символы национальных алфавитов; псевдографику; научные символы.

Дано: символ i имеет в таблице ASCII номер 105. Какое слово зашифровано в следующей последовательности символов: 108 105 110 107, учитывая что нумерация идет по алфавиту.

LINK

# Измерение информации

Человек для общего понимания всеми придумывал единицы измерения:

1. Материя – измеряется в количестве вещества (обычно определяемого массой вещества). Яблоко = 200 гр.; Атом = 1 \* 10 -20  кг. ; Земля = 6 \* 10 26 кг.
2. Энергия – тепловая, механическая, электрическая, атомная – измеряется в количестве энергии – Ватты, джоули, Килокалории. Взмах крыла мухи несет энергию 10 -4 Дж.; Вылет пули из ружья – 10 +2 Дж.;Взрыв атомной бомбы = 10 20 Дж.
3. Информация – уменьшение неопределенности нашего знания об исследуемом объекте в 2 раза несет 1 бит информации. Неопределенность – количество возможных исходов (результатов) данного события.

Дано: на экзамене могут быть оценки от 2 до 5. Студент получил оценку хорошо. Разработать алгоритм измерения кол-ва информации для определения его оценки, задавая вопросы так, чтобы на каждом шаге неопределенность уменьшалась в 2 раза.

## Формула Хартли

2 i = N

N – кол-во возможных исходов;

i – кол-во информации в сообщении в битах

i = log 2 (N)

16 стелажей, 8 полок, кНига на 4 стелаже, 3 полке – 7 бит.

В этой формуле N – кол-во равновероятных исходов. Если события равновероятны, то вероятности наступления каждого равны. Если события не равновероятные, то вероятность наступления такого события P = K/N, где K – кол-во событий, относящихся к рассмотрения, а N – общее кол-во событий.

Для достоверного события P = 1, для невозможного события P = 0.

Чем меньше вероятность события, тем больше оно информации несет.

Дано: в корзине черные и белые шары. 18 черных. Сообщение о том, что достали белый шар несет 2 бита информации. Определить сколько шаров в корзине.

Пусть x – кол-во шаров в корзине.

I = log 2 (1/P) = 2 =>

1/P = 4 =>

P = 0.25 =>

N = 18 / (1-P) = 24

Если события не равновероятные, то применяется формула Шеннона.

Формула предполагает, что любое I k = -1/N \* log 2 (1/N) – кол-во информации, полученное от k-того исхода.

Дано: в корзине 32 шара, красные и белые. Из них 4 красные. Сколько информации в сообщении, что достали красный шар? Сколько информации в сообщении, что достали любой шар. - **ДЗ**

## Алфавитный подход к измерению информации

### Кибернетика

- Наука, которая занимается сложными объектами, в т.ч. живыми организмами, воспринимая их как систему. Но, кибернетику интересует не содержание объекта, а взаимодействие объектов между собой. Вывод: Кибернетика – наука о процессах управления и связях в живых и неживых системах. Объект представляется «Черным ящиком», который имеет входные и выходные сигналы. Сигналы представлены в виде последовательности символов.

Дано: Русский язык, алфавит – 54 символа (N). Кол-во символов в алфавите – его мощность. Определить, какое кол-во информации несет 1 символ, если считать, что появление в тексте любого символа равновероятно.

N = 2 I I = 5,755.

Страница 50 строк по 60 символов. Какое кол-во информации содержит такая строка? 17 265 бит.

Английский алфавит, 47 символов. 1 символ = 5,555. Страница = 16 665

Вывод: кол-во информации в символьной системе зависит от мощности алфавита и размерности текста => нельзя сравнивать информационное содержание только по размеру текста.

Задача: Дано 2 сообщения, равное кол-во символов. В первом сообщении кол-во информации в 1.5 раза больше чем во втором. Мощность обоих алфавитов не превышает 10 символов в каждом алфавите. При этом каждый символ несет целое число бит информации. Найти мощности алфавитов.

# Системы контроля версий

## Version Control System (VCS)

1. Отслеживание всех изменений внесенных в программный код.
2. Хранят историю изменений программного кода.
3. Реализуют методы и алгоритмы просмотра и восстановления сохраненного состояния.
4. Обеспечивают функции возвращения кода к рабочему состоянию на любом этапе разработки.
5. Организуют локальный или удаленный репозиторий.

СКВ обеспечивают работу нескольких программистов над одним проектом.

## Типы СКВ

1. Локальный – копирование файлов в отдельную директорию. Недостатки:
   1. Капец как ненадежно.
   2. Нельзя работать вместе.
2. Централизованные СКВ – при такой схеме единственный сервер содержит все версии файлов. Разработчики получают файлы из централизованного репозитория, там же из сохраняют. Недостатки:
   1. потеря всех данных, если сервер выходит из строя.
   2. Затруднена одновременная работа над одним файлом для нескольких разработчиков.
3. Распределенные СКВ – дополняют схему централизованной тем, что дают возможность хранить полноценную копию репозитория на локальной машине. Разработчики скачивают снимок всех файлов, полностью копируя репозиторий. У каждого разработчика есть копия исходного кода и копия внесенных изменений. Репозиторий любого разработчика копируется на другой сервер если рабочий выходит из строя. Как следствие, возможна параллельная работа над одним и тем же файлом нескольких разработчиков. Так же, это позволяет одновременно взаимодействовать с несколькими удаленными репозиториями. Облегчается работа над несколькими проектами сразу.

## Виды СКВ

1. Subversion – старость.
2. Git – Разработал Линус Торвальдс, как систему для разработки ядра линукса.

В Git хранение данных осуществляется в виде набора снимков файловой системы. При сохранении состояния проекта Git запоминает каждый файл и сохраняет ссылку на этот снимок. Побочным эффектом является механизм резервного копирования.

# Творческое задание

1. Какие технологии сейчас исследуются и испытываются? Какие являются наиболее перспективными? Почему?

Ответ:

Чиплеты - микросхема, специально разработанная для совместной работы с другими себе подобными. Несколько чиплетов формируют одну более большую и сложную микросхему. Например такую, как центральный или графический процессор.

Обычные монолитные микросхемы имеют собственную подложку — тонкую пластину, посредством которой выводы чипа соединяются с массивом шаров из припоя. В случае с чиплетной микросхемой подложка для всех чиплетов используется общая. Сами чиплеты подключаются между собой соединениями внутри подложки, и только потом общие выводы с нескольких чипов соединяются с массивом алюминиевых шаров.

Преимущества чиплетов:

* Монолитные кристаллы для центральных процессоров и видеокарт достаточно крупные, и при использовании кремниевых пластин стандартного типоразмера 300 мм остается достаточно много неиспользованного места, которое потом утилизируется. При использовании более мелких кристаллов чиплетов можно задействовать больше полезной площади пластины и уменьшить количество утилизируемого кремния.
* В одном продукте можно использовать чиплеты, произведенные по разным техпроцессам. Благодаря этому для некоторых мало потребляющих узлов чипа можно использовать более зрелые и «толстые» техпроцессы, у которых меньше уровень брака и дешевле производство пластины. Это положительно сказывается как на затратах производителя, так и на стоимости конечного продукта.
* Чиплеты позволяют гибко конфигурировать готовый продукт, в отличие от монолитных кристаллов. В случае необходимости, компания-производитель может доработать один или несколько из чиплетов уже выпускаемого продукта, при этом не затрагивая остальные. Затрат на разработку такого продукта у производителя будет намного меньше, чем при планировании переделки уже существующего монолитного чипа и выпуска его модификации.

Я считаю такую технология наиболее перспективной в текущей ситуации, так как применение чиплетов не требует глобального изменения производственных линий, позволяет получать из одинаковых компонентов продукты разного ценового сегмента и самое главное – позволяет преодолеть закон Мура, гласящий, что каждые 2 года мощность процессоров увеличивается в 2 раза. Наблюдая за процессорами последние пару лет можно было сделать вывод, что данный закон почти перестал работать, ведь до выхода процессоров AMD Ryzen прирост составлял не более 25%-30% за 2 года. С выходом первых Ryzen, основанных на чиплетной архитектуре наконец-то начались значимые скачки в производительности, данная компоновка кристалла так себя зарекомендовала, что даже компания-конкурент Intel в недавнем времени начала переход к чиплетной компоновке процессоров.

1. Какую ЭВМ разрабатывали бы вы? Какие технологии использовали бы? Какие идеи есть?

Исходя из изложенной выше информации, я бы предположил, что в будущем ЭВМ будут основываться на чиплетной компоновке, но в немного другой форме. Если сейчас чиплеты выполняют одинаковые функции и их главное различие – соотношение мощность/энергопотребление, то в будущем можно предположить, что разные чиплеты будут выполнять разный функционал, например ускорение отрисовки изображения (замена видеокарты) или заменят собой оперативную память. Такая компоновка имеет ряд преимуществ перед современными системами и уже начинает внедрятся в потребительские компьютеры, например в новых чипах Apple Silicon все компоненты, такие как процессорные ядра, видеоядра, оперативная память, оба моста, множество кодеков и декодеров. Размещение разных блоков на одной подложке рядом друг с другом позволяет им быстрее общаться и увеличить производительность каждого. Но на данный момент, такую компоновку использует только Apple. Может показаться, что интегрированная графика в современных процессорах тоже подходит под данное определение, но если копнуть глубже, то выясняется, что это самое видеоядро общается в памятью с помощью процессора, а не самостоятельно, так как такой подход был разработан до создания чиплетной компоновки. Из минусов такого подхода можно выделить низкую производительность современных графических ядер в процессорах

1. Что будет дальше?

|  |  |  |  |
| --- | --- | --- | --- |
| **Года** | **Компьютерные технологии** | **Задачи** | **ПО** |
| 1996 год – Н.В. | * Элементная база – большие интегральные схемы. * Появление новых компьютеров – сетевых компьютеров. * Увеличение быстродействия до десятков триллионов операция в секунду. * Уменьшение габаритов до уровня наладонного прибора. * Уменьшение стоимости в 1000 раз. * Увеличение оперативной памяти за счет внешних выносных устройств. * Появление новых каналов связи: Оптоволоконная связь; Сотовая связь; Беспроводная связь. * Устройства голосового управления, Аватары.   Среды, которые позволяют визуальное общение человека и системы? | 1. Разработка огромных хранилищ данных, у которых появились новые функции: не только хранение, а поиск, обработка данных, анализ связей и отношений между объектами 2. Информационная разведка. 3. Задачи экономического развития. 4. Задачи моделирования. | * Появление новой оболочки языков – Искусственные языки – Языки, основанные на копировании языка человеческого общения. * Появление современных программных сред. Их действие основано на: преобразовании задачи в проект, где каждый шаг представлен в виде имиджа (формулы, словесного описания, графического изображения, нотных знаков, рисунков и других средств) * Появление интеллектуальных программных систем: 1. Системы анализа текстов для выявления резюме – наиболее важные и краткие данные об объекте. Цель – Автоматический выбор данных по заданным параметрам. 2. Генераторы технических решений. Определение направления развития технических решения, научной мысли, Бизнес-идеи (Мозговой штурм) 3. Системы моделирования: 3.1 Системы ситуационного моделирования основаны на принципе «если-то»; 3.2 Системы имитационного моделирования, которые исследуют реальные системы, используя их описание в виде систем уравнений (формулы, музыка) 3.3 Когнитивные модели – моделирование, которое происходит в программных средах. Позволяет выстраивать прогнозы решений по курсу валют, по климату, по производственному процессу. Строятся на основе больших объемов данных. 4. Этот этап характеризуется исчезновением посредников между человеком и компьютером. Программисты не исчезают, системотехники не исчезают, они занимаются разработкой ОС, новых ЯП, разработкой программных сред. |

# Характеристики КТ и ИТ на современном этапе

## Основные тезисы

1. Заказчик всегда знает чего хочет, но не знает, как это сделать. Любая постановка задачи должна быть определена в общепринятых терминах, понятных как заказчику, так и разработчику.
2. Реальность всегда сложнее абстракции. Любая задача, проект проходит стадию моделирования. В создании модели должны участвовать заказчик и исполнитель. Модель должна оперативно изменяться по структуре, параметрам и элементам связи.
3. Результат получается там, где получено хорошее, наглядное изображение модели. Проект должен содержать раздел визуализации структурной, символьной, графической, звуковой и других возможных средств.
4. Проклятие размерности. Заказчик в процессе реализации нагружает модель все более сложными дополнительными функциями. Задача разработчика отсечь те функции, которые не ухудшают работу системы, но если функция необходима, то разработчик дополняет ТЗ.
5. Немонотонность человеческой логики. Решение состоит в том, что в алгоритмах моделей должны быть зашиты заранее предусмотренные действия (например, Система СТОП)
6. В реальных ситуациях часто бывает так, что задача когда-то кем-то уже решалась. Наши действия: Решение найти, проанализировать и применить или отбросить в нашем проекте.
7. Основные тенденции работы с информацией:
   * Любая информация по проекту представляется в структурированном виде: графики, диаграммы, знаки, фото, видео и т.д. Но этот вид должен предусматривать возможность автоматизацию обработки информации.
   * Интуитивно понятный интерфейс.
   * Любая информация должна предоставлять возможность иметь тот набор средств, который ее обрабатывает.
   * Любая информация, информационная среда должна содержать средство помощи пользователю.
8. Типы информации:
   * Структурная информация. Ею обладают все материальные объекты. В природе она существует в потенциальной форме и является устойчивым свойством материи. Накопление ее повышает уровень организации материи и, следовательно изменяет его структурно. Вывод: структурная информация описывает устройство материи.
   * Оперативная информация. Используется в целях управления и познания, то есть описывает состояние материи.

# Классификация современных компьютеров

1. Суперкомпьютеры – компьютер очень высокой мощности, единичного производства для решения уникальных задач. Не экспортируется, решают гос. Задачи обороны, хранения гос. Тайны, кодирования информации, не подключаются к мировой паутине. (США, Россия, Германия, Китай, Индия, Израэль, Япония)
2. Мейнфрейм – компьютер высокой мощности для интенсивных вычислений, связанных с хранением, поиском, обработкой информации. Для многопользовательского режима связаны между собою в сеть, обычно отраслевого назначения, высокой стоимости. В России владеют такими Сбербанк, МЧС + ГИБДД.
3. Рабочие станции - Мощный компьютер для отдельного пользователя для выполнения узкого круга задач. Стоит дорого.
4. ПК – массовый, унифицированный, для широкого круга, имеющий функции ввода/вывода, коммуникации, манипуляции данными, их хранения и обработки.
5. Ноутбуки – все функции ПК, персональное пользование и минимальный вес (менее 3 кг)
6. PDA – карманный или наладонный прибор, распознающий рукописный ввод, мощное средство коммуникации, предназначен для решения оперативных задач.

# Развитие человека. Цивилизации. Технологий.

Для того, чтобы начинать процессы автоматизации и понимания надо начинать с:

1. Изучение закономерностей развития человеческой цивилизации.
2. Поймать тенденции во всех направлениях развития.
3. Экстраполировать изученное на современный этап и на будущий.

Зачем: выстроить новый этап развития во всех направлениях.

|  |  |  |  |
| --- | --- | --- | --- |
| **Этап** | **Характеристики** | **Выводы** | **Выход** |
| 1 Этап.  Дикое общество | 1. Отсутствие безопасности. (охота, борьба семей, дикая природа и ее факторы, болезни, голод) | Природа могущественнее человека, сила и скорость животных выше человеческой, естественный отбор => человеческая популяция не растет | Новые идеи, мозг человека.  Появление первых идей, таких как   * **накопление запасов**, **самое безопасное место – дом,** * звери боятся огня * Поддержка огня -> кооперация * Необходимо организовать проживание так, чтобы все было рядом (огонь, еда, вода и т.д.) -> рядом с домом животные, пахотные земли, плодовые растения.   Все это вынуждало человека вынуждало человека организовать постоянный дом. |
| 2 Этап.  Оседлость |  |  | Первая революция – аграрная революция (**дичь** + растения рядом с домом)  Спираль прекращения голода => Каждый человек был обеспечен едой => Гарантия жизни => Иногда неурожай (истощение земли, кражи соседями) => Полив посадок, выпалка грядок, защита от соседей для нивелирования неурожаев => Измерения, вычисления, наблюдения для обеспечения равенства между людьми => Появление права, Разделение участков (Владение землей) => Появление понятия собственности => Появление помощников (наемный труд, рабовладение) |
| 3 этап.  Промышленное общество | * Специализация * Одинаковые орудия труда * Стабилизация получения первичных надобностей   Снижение рисков 2 этапа => сосредоточение на человеке, кто делает орудия труда => кто делает машины, делающие орудия труда  Основы => промышленность => машины  “+”: увеличивается скорость, не устают, улучшается качество и одинаковость результата, размножать рез-ты, продавать | Взаимодействие разных людей с разными профессиями. Появление городов: критерии:   * Колодец * Площадь * Охранная башня * Школы(образование) * Церковь(мораль) * Ратуша(закон) * Крепостная стена   Наука => появление интеллектуального труда:   * Криптография * Письменные языки * Появление мер * Понимание понятия энергии – связь с материей | Идея- человек изобретает машину, которая делает другие машины => конвейеры => индустриальная рев-ция - общество стремится к максимальной эффективности=> стандартизация (одинаковость) Человек хочет увеличить КПД (уменьшить затраты, рез-т увеличить):   * Узкая специализация * Массовое производство * Культура * Потребление * Образование   1 путь: развитие, искусство, технологии  2 путь: разнообразие товаров, переработка ресурсов, индустриальное общество развивается, промышленными гигантами необходимо управлять, усложнение и удлинение линий конвейеров, централизация власти => системам управления пирамида  Пирамидное управление вызывало:   * Рост управляющих различных назначений * Основание пирамиды не может прокормить пирамиду => коллапс, разрушающий пирамиду   Выход: ± объединение пирамид, конфликт(война, укрепление, энергия, информация), 2 образование: характеристика: потоковое, массовое, узкоспециализированное(уменьшение количества навыков специалиста, более глубокие знания конкретного предмета) => ВУЗы обучают усилителям 3. Искусство: реализм, понимание мира как машины, узнаваемость изображенного 4. Гуманитарные науки: изучение человека, стагнация общества, человек обществу не интересен, человек не познан!!1!!. Вывод: не учитывать человека нельзя – в его руках головах: сложные системы (вооружение), производство, армия, города, население. Вывод: гаджеты, устройства, технологии могут быть использованы одним в ущерб многим. |
| 1. Индустриальное общество | Новые научные исследования ( антибиотики => увеличение длительности жизни в 3 раза, снижение рисков проживания. Появление контрацептивов => регуляция населения. Появление мощных диагностирующих устройств: УЗИ, МРТ, с чем связанна ранняя постановка диагноза, увеличение человеческой жизни) | Использование массового производства. Появление мощной автоматизации. Техническое развитие человека. Так ка слишком высоки темпы извлечения ресурсов цивилизации. Повышение мобильности услуг и людей, результатов производства. Взрыв связей из-за |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Теория алгоритмов

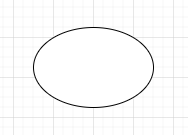
## Формы алгоритмов

1. Словесная форма
2. Блок-схема
3. Программный код

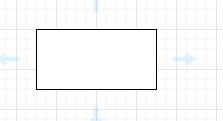
Блок схемы алгоритмов (ГОСТ 19.701-90)

## Блоки блок-схем:

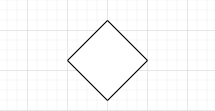
1. Блок начала и конца алгоритма – Овал



1. Блок действия – Прямоугольник

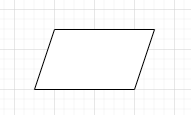


1. Блок условия – Ромб (для while, if и тд)

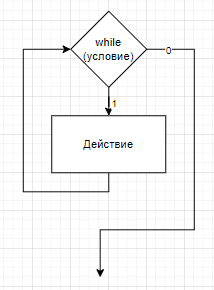


Оформление – две стрелочки от двух углов, с подписью 1(условие выполнено) и 0(условие не выполнено). 1 всегда налево, 0 направо. После ветви идет выполнение действия. У оператора ветви сходятся в одной точке независимо от выполненной ветви. В c++ ветка 1 (то) не может быть пустой.

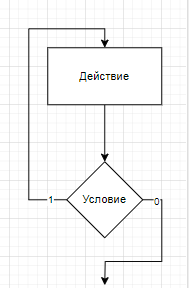
1. Блок вывода – ввод и вывод данных



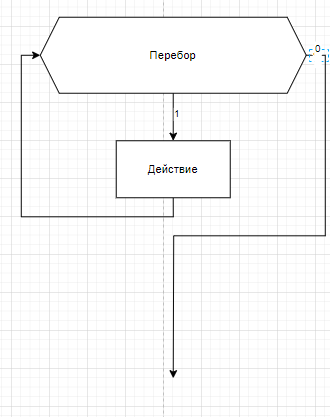
1. Оператор while



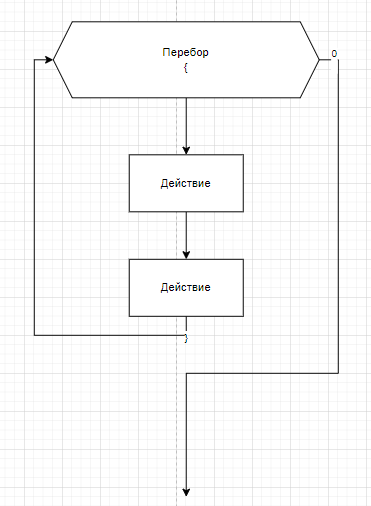
1. Цикл с постусловием



1. Цикл с параметром

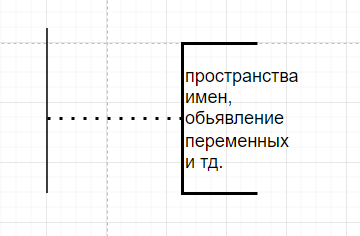


Правило: Несколько действий обрамляются фигурными скобками ( «{}» )



Есть второй тип цикла с параметром – цикл foreach. Для обработки коллекций, строк и тд.

1. Сноска

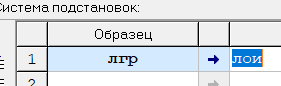


## Модель алгоритма. Нормальные алгоритмы Маркова.

Модель алгоритмов Маркова преобразует исходные слова с помощью заданных правил, которые могут заменять часть исходной строки. Слово – строка из символов. Кроме правил в алгоритмах Маркова используются приоритеты их применения.

## Принципы машины Маркова

1. Правила находят первое вхождение свое в исходной строке и заменяет его на вторую часть правила



1. Правила выполняются по порядку их номеров



1. Если правило не применят
2. Если в исходной строке, то алгоритм переходит к проверке следующего по порядку правила
3. Если правило выполнено, то алгоритм возвращается к проверке правил, снов аначиная с начала списка.
4. Если ни одно правило нельзя применить, то алгоритм завершается.

## Подстановки и примечания

Виды подстановок:

1. Терминальные – принудительное завершение программы после выполнения подстановки (\* |-> <пусто>)
2. Не терминальные (\*a -> aa\*)

Примечания:

1. Если в левой части правила стоит пустая строка, то к слову в начало добавляется строка из правой части правила
2. Если правая часть правила пустая, то подстрока исчезает из слова.