

Типы данных

Занятие #1
по курсу Основы языка С

План лекций

- ◆ Типы данных (data types)
- ◆ Функции и переменные (function and variables)
`scanf("%d", &x);`
- ◆ struct, union
- ◆ if, логические операции (logical operations)
- ◆ циклы (loops)
- ◆ массивы и указатели (arrays and pointers)
работа с динамической памятью (memory allocation)
- ◆ строки (strings)
- ◆ сортировки + qsort
- ◆ структуры данных: списки, очереди, деревья, графы
- ◆ динамическое программирование

Стандарты

- ◆ C89
- ◆ C99
- ◆ C11
- ◆ C18 (bugfix)
- ◆ диалекты и расширения
 - gcc
 - clang
 - intel
- ◆ Кто прав? Только стандарт.

Hello, world

```
#include <stdio.h>
```

Включение информации о стандартной библиотеке ввода-вывода
(без этой строки printf не работает)

```
int main ( )  
{
```

С функции **main** начинается выполнение программы

```
printf ("Hello, world!\n");
```

Печать строки

```
return 0;
```

```
}
```

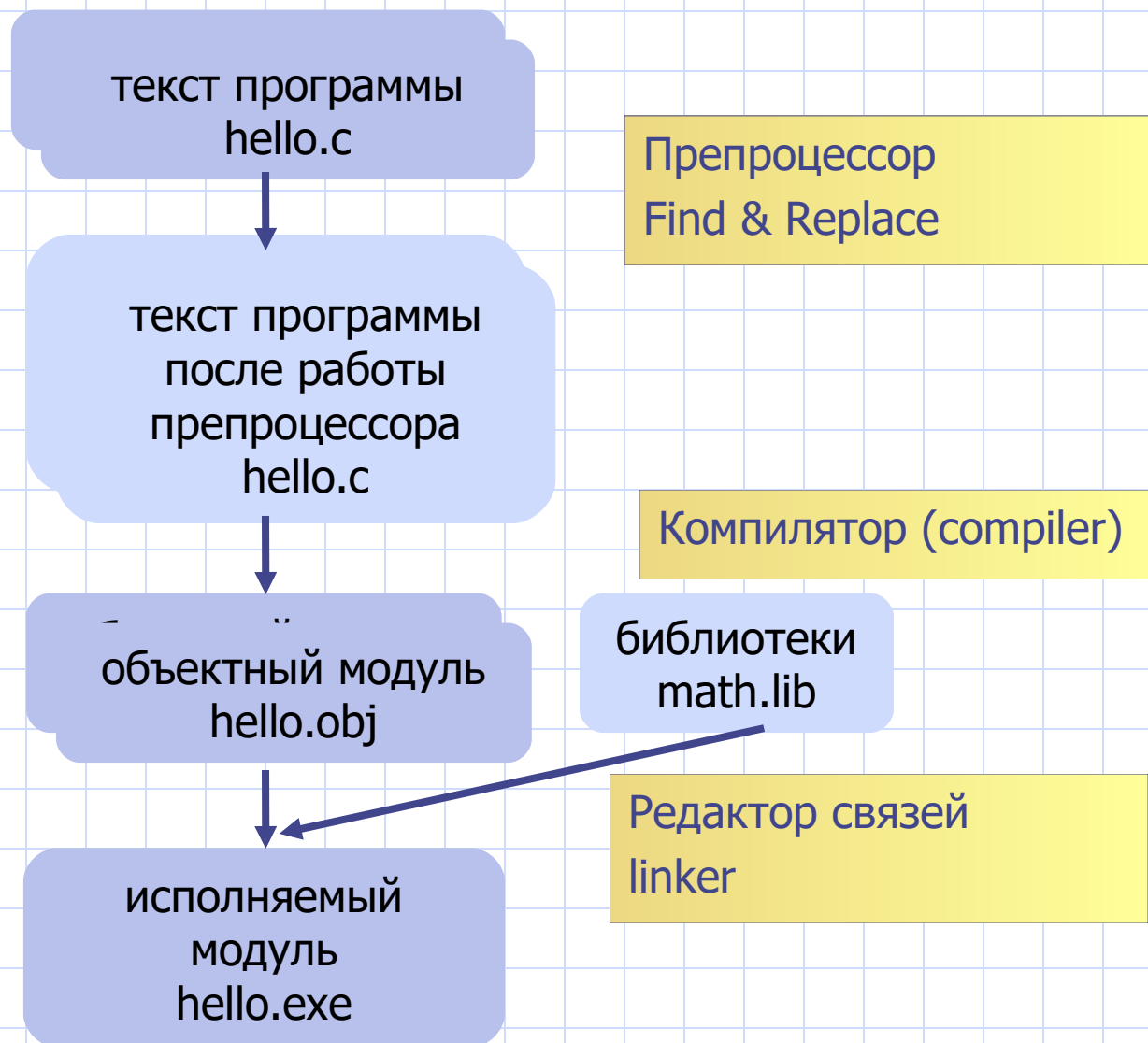
Возвращаем целое число
(обычно код ошибки,
0 - Success)

Escape sequence

```
printf ("Hello, \"Titanic\"!\n");
```

- ◆ Специальные символы
- ◆ **\n** – newline (курсор в начало новой строки)
- ◆ **\r** – return (курсор в начало текущей строки)
- ◆ **\t** – TAB (табуляция, курсор на позицию кратную 8 символам)
- ◆ **\b** – backspace (курсор назад)
- ◆ **\"** – кавычки "
- ◆ **\'** – одинарная кавычка '
- ◆ **** – \
- ◆ **\a** – alarm (звуковой сигнал)

От текста к бинарному коду



Компиляция и запуск программы

- ◆ В командной строке
- ◆ компиляция
`gcc -Wall -o hello.exe hello.c`
- ◆ запуск программы
`./hello.exe`
- ◆ с исполняемым модулем по умолчанию (*a.out* или *a.exe*)
`gcc -Wall -Wextra hello.c`
`./a.out`
- ◆ `man 3 printf`

Печать десятичного числа

Печать значения переменной
в виде *десятичного* числа

```
#include <stdio.h>
```

```
int main ( )  
{
```

Можно написать декларацию с явной
инициализацией
int x = 2;

```
int x;  
x = 2;
```

Decimal is 2


```
printf ("Decimal is %d \n", x);
```

```
return 0;
```

```
}
```

d - десятичное decimal
o - восьмеричное octal
x - шестнадцатеричное hex

Переменные

- ◆ Переменная - именованная область памяти, в которой хранится значение 
- ◆ А и а - различаются case sensitive **X**
- ◆ Имена переменных (идентификаторы)
 - a-z A-Z 0-9 _
 - не может начинаться с цифры
- ◆ что не идентификатор?
 - x12 X12 12X _12
 - x_12 _x12 12 _
- ◆ Почему идентификатор не может начинаться с цифры? 12L, 3f, 3e8

=

◆ Оператор присваивания (assignment)

- `x = 7;` `// ok`
- `7 = x;` `// Error`

x 

- Значение выражение справа `rvalue`
- Область памяти для хранения (переменная) `lvalue`

- `res = x + y;`
 - взять значения `x` и `y` (`load`)
 - вычислить `x + y` (`evaluate`)
 - результат записать (`save`) в `res`

Печать двух переменных

```
#include <stdio.h>
```

```
int main ( )
```

```
{
```

```
    int x = 10, y = -7;
```

```
    printf ("%d plus %d is %d\n",  
            x,          y,    x + y);
```

```
    return 0;
```

```
}
```

10 plus -7 is 3

Печать многих переменных

```
#include <stdio.h>
```

```
int main ( ) {
```

```
    int x, y, res;
```

```
    x = 10;
```

```
    y = -7;
```

```
    res = x + y;
```

```
    printf ("%d plus %d is %d\n", x, y, res);
```

```
    return 0;
```

```
}
```

10 plus -7 is 3

Арифметические операторы

- ◆ + сложение
- ◆ - вычитание
- ◆ * умножение
- ◆ / деление
- ◆ % деление по модулю
(остаток от деления)
9%5 даст 4

$(1 + 2) * 3$
 $1 + 2 * 3$

- ◆ Приоритет операций, скобки
- ◆ Слева направо или справа налево?

Краткая форма

◆ Сочетание = и операций *= -= /= ...

◆ $x = x + 2;$

◆ $x += 2;$

◆ $x = x + 1;$
 $x += 1;$

$x ++;$	$x--;$	// результат старый
$++ x;$	$--x;$	// результат новый

Прочитать десятичное число

```
#include <stdio.h>
```

```
int main ( )
```

```
{
```

```
    int x, y;
```

```
    scanf ("%d", &x );
```

```
    scanf ("%d", &y );
```

```
    printf ("Div is %d \n", x/y);
```

```
    return 0;
```

```
}
```

Подключена стандартная библиотека ввода-вывода
(без этой строки scanf и printf не работают)

Считать десятичное число в переменную x

d - десятичное decimal
o - восьмеричное octal
x - шестнадцатеричное hex

Целочисленные типы данных

- ◆ char — самый короткий [-128..+127]
- ◆ short *int* [-32768 .. +32767]
- ◆ int
- ◆ long *int*
- ◆ long long *int*
- ◆ спецификаторы
 - unsigned, signed
- ◆ С плавающей точкой 3.14
 - float
 - double

Преобразование типа (cast)

- ◆ неявное преобразование
implicit cast (warning)

```
int x = 5.0;
```

- ◆ явное преобразование (explicit)
(*type*)

```
x = (int) 5.0;
```

```
x = (int) (y + z);
```

```
x = y + (int) z;
```

- ◆ Отбрасываем дробную часть

- $12.56 \rightarrow 12$

- $-12.56 \rightarrow -12$

Преобразование типов

порядок выполнения

типы операндов

тип результата

7 / 3

```
double x = 7 / 3;  
           int  int
```

2.0 ← int 2

```
double x = 7 / 3.0 ;    // 3.    3f
```

```
double x = 7 / (double) 3;
```

Преобразование типов

- ◆ от простого к сложному: `int + double`
- ◆ от `signed` к `unsigned`: `s + u = u`
- ◆ от короткого к длинному `int + long`
- ◆
`int x = 12;`
`long int y = - 4;`
`unsigned int a = 3;`
`float k = 1.5;`
- ◆ `x + k` `// int + float = float`
- ◆ `x + y` `// int + long = long`
- ◆ `x + a` `// signed unsigned int`
- ◆ `y + a` `// unsigned long`

Константы

- ◆ 12, 12L, 12u, 3.14, 3.14f, 1e9, 'z'
 - "Hello"
 - 012, 0x12 – octal and hex

Форматы printf и scanf

◆ %width.precision length **format**

◆ Format:

d dec, o octal, x hex

u unsigned dec

f, g, e — double

%02d:%02d
hh:mm

◆ %6d — печатать в 6 позиций

◆ %.2f — печатать 2 знака после запятой

◆ %06.2f — печатать 2 знака после запятой
в поле на 6 позиций с ведущими нулями

◆ l или ll — модификаторы long и long long
%lld %lu %lf (double для scanf)

◆ h или hh — short и char %hf

Позиционные системы счисления

◆ Десятичная $615_{10} = 6 \cdot 10^2 + 1 \cdot 10^1 + 5 \cdot 10^0$

10^3	10^2	10^1	10^0
0	6	1	5

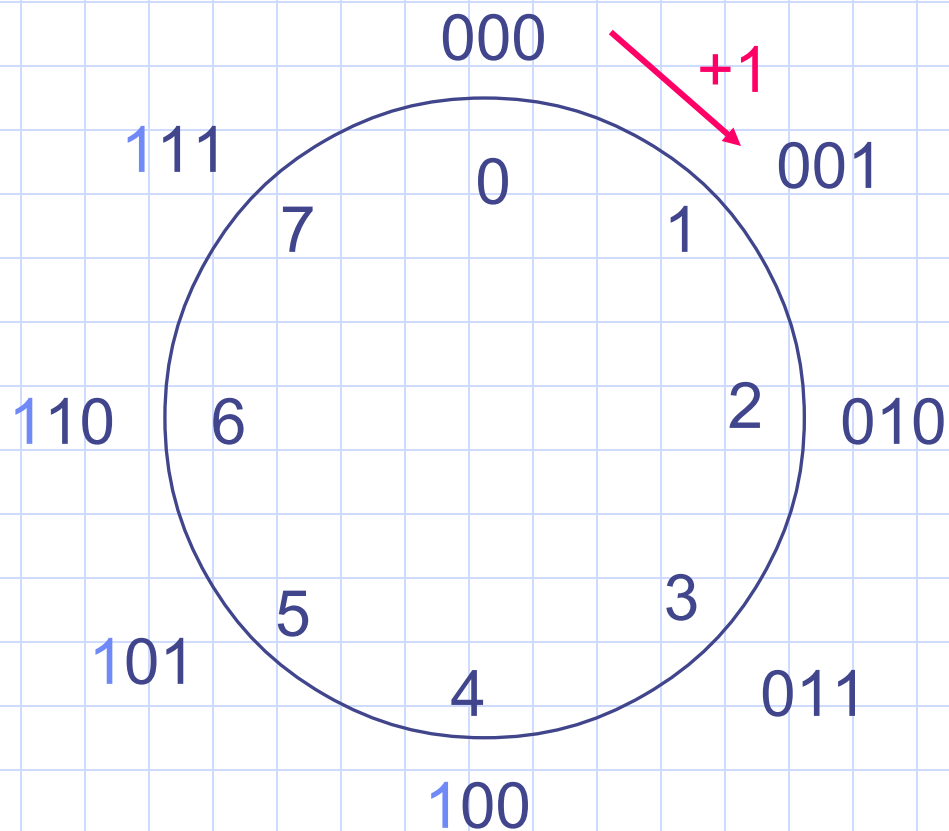
◆ Двоичная $1101_2 = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$

2^3	2^2	2^1	2^0
8	4	2	1
1	1	0	1

$1 \cdot 8 + 1 \cdot 4 + 0 \cdot 2 + 1 \cdot 1 = 13_{10}$

$1101_2 = 13_{10}$

Диапазон положительных чисел



- ◆ 3 битные числа
- ◆ $2^3 = 8$ чисел
- ◆ 0 ... 7
- ◆ $7 + 2 = 1$
- ◆ unsigned

Представление отрицательных чисел

◆ $x + (-x) = 0$ в побитовом сложении

◆ знаковый бит (signed bit)

◆ Прямой код (старший бит 1)

3: 0011

-3: 1011

◆ Обратный код ($\sim x$)

3: 0011

-3: 1100

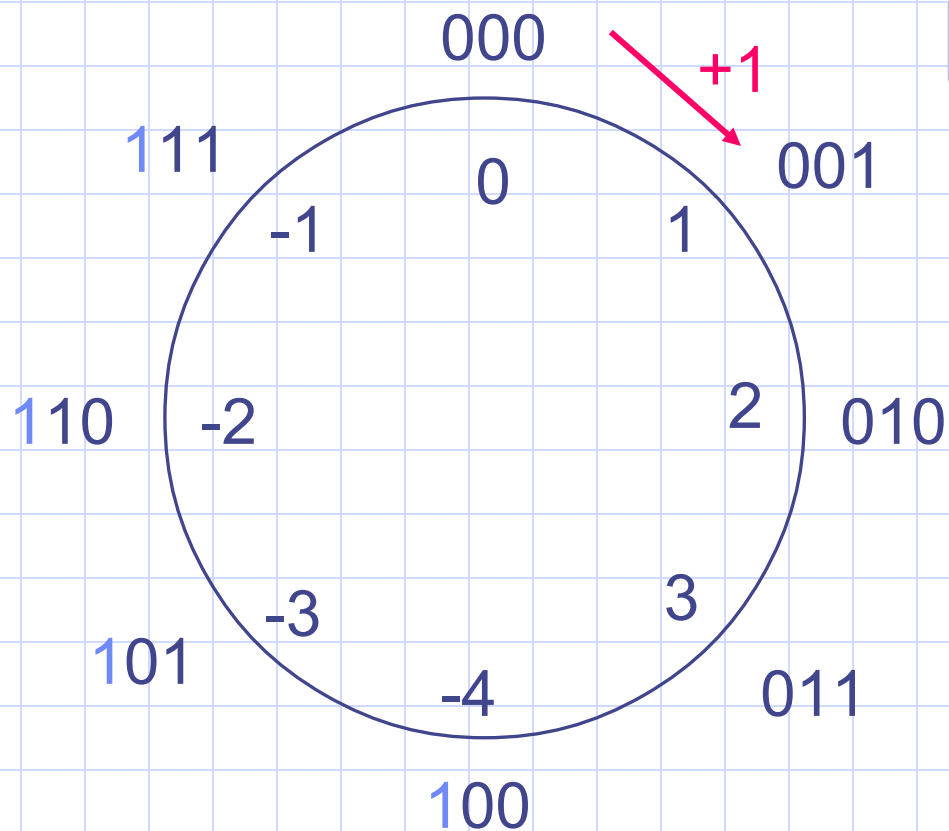
◆ Дополнительный код: $x + (-x) = 0$

3: 0011

-3: 1101 (обратный +1)

Σ : 10000

Диапазон знаковых чисел



дополнительный код

- ◆ 3 битные числа
- ◆ $2^3 = 8$ чисел
- ◆ - 4 ... 3
- ◆ $3 + 1 = -4$
- ◆ signed

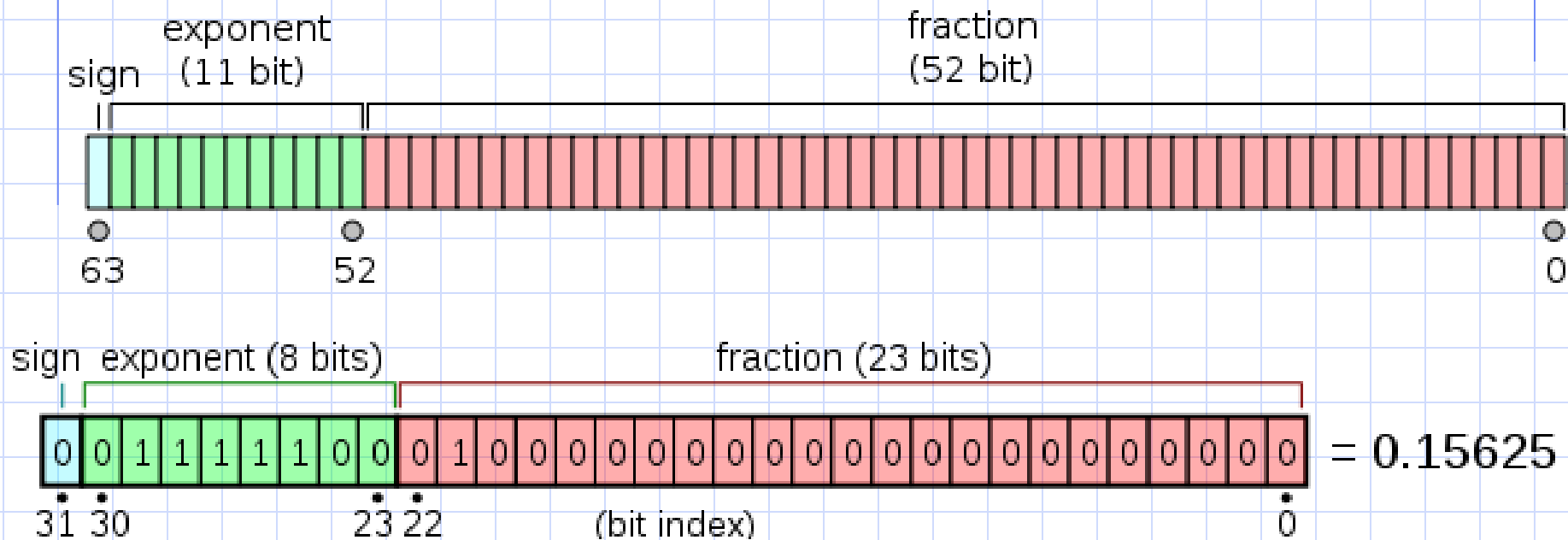
Представление дробных чисел

◆ **fixed-point number (в C нет)**

- считаем деньги
(тут запятая всегда фиксирована)
- SQL
- обработка сигналов (звук и изображение)
часто в fixed-point
- точные вычисления

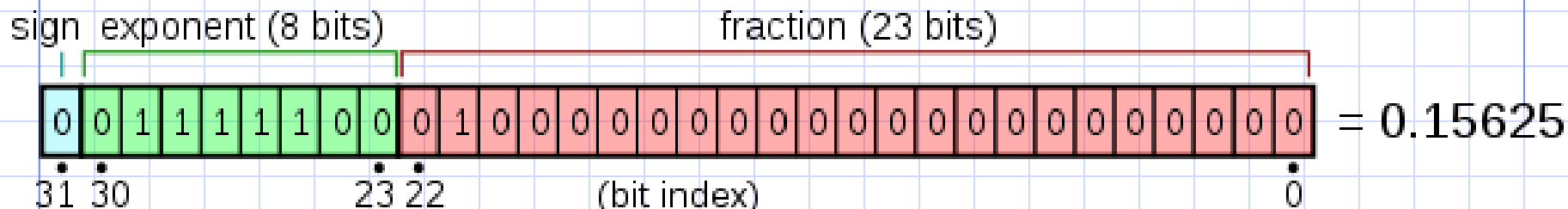
Floating-point number

$$\diamond (-1)^{\text{sign}} \cdot 2^{\text{exponent} - \text{exponent_bias}} \cdot 1.\text{mantissa}$$



Floating-point number bin to dec

$$\diamond (-1)^{\text{sign}} \cdot 2^{\text{exponent} - \text{exponent_bias}} \cdot 1.\text{mantissa}$$



$$\begin{aligned} & (-1)^0 \cdot 2^{124 - 127} \cdot 1.25 \\ &= 2^{-3} \cdot 1.25 \\ &= 0.15625 \end{aligned}$$

bit 23 = 1 (неявно задан)

bit 22 = 0.5

bit 21 = 0.25

bit 20 = 0.125

bit 19 = 0.0625

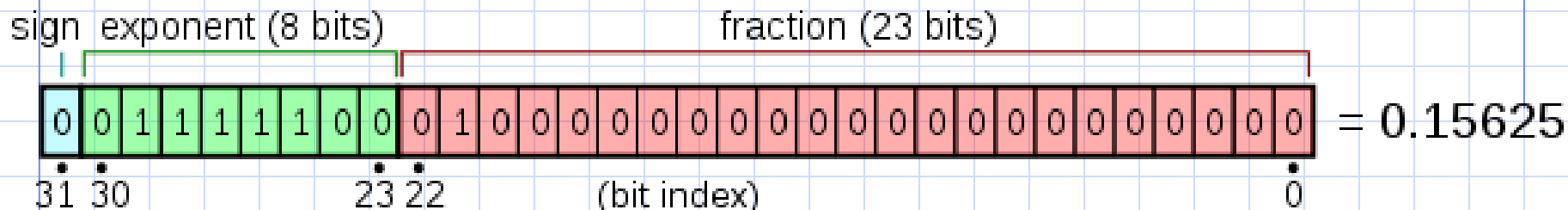
.

bit 0 =

0.00000011920928955078125

Задача bin to dec

$$\diamond (-1)^{\text{sign}} \cdot 2^{\text{exponent} - \text{exponent_bias}} \cdot 1.\text{mantissa}$$



bit 23 = 1 (неявно задан)

bit 22 = 0.5

bit 21 = 0.25

bit 20 = 0.125

bit 19 = 0.0625

.

bit 0 =

0.00000011920928955078125

41c8 0000₁₆

Решение

◆ $41c8\ 0000_{16} =$

$0100\ 0001\ 1100\ 1000\ 0000\ 0000\ 0000\ 0000_2$

◆ Sign bit: 0

◆ Exponent: $1000\ 0011_2 = 83_{16} = 131_{10}$
 $131 - 127 = 4$

◆ Significant (с дописанным неявным битом):
 $1100\ 1000\ 0000\ 0000\ 0000\ 0000_2 = c80000_{16}$
 $= 1 + \frac{1}{2} + \frac{1}{16} = 1 + 0.5 + 0.0625 = 1.5625$

◆ $(-1)^0 \cdot 2^{131-127} \cdot (1 + \frac{1}{2} + \frac{1}{16}) = 2^4 + 2^3 + 2^0 = 25$

12.357 dec to bin

◆ $12.357 = 12 + 0.357 = 1100_2 + (.011)_2$

■ $0.357 * 2 = 0.714 = 0 + 0.714$

$0.714 * 2 = 1.428 = 1 + 0.428$

$0.428 * 2 = 0.856 = 0 + 0.856$

◆ $(12.357)_{10} = (1100.011)_2 =$

$(1.100011)_2 * 2^3 = (1.100011)_2 * 2^{130-127}$

◆ Sign bit: 0

◆ Exponent: $1000\ 0010_2 = 82_{16} = 130_{10}$

◆ Significand: $1100\ 0110\ 0000\ 0000\ 0000\ 0000_2$

◆ $0100\ 0001\ 0100\ 0110\ 0000\ 0000\ 0000\ 0000_2$

Периодическая дробь 0.1


$$0.1 * 2 = 0.2 = 0 + 0.2$$

$$0.2 * 2 = 0.4 = 0 + 0.4$$

$$0.4 * 2 = 0.8 = 0 + 0.8$$

$$0.8 * 2 = 1.6 = 1 + 0.6$$

$$0.6 * 2 = 1.2 = 1 + 0.2$$

$$0.2 * 2 = 0.4 = 0 + 0.4$$

$$0.4 * 2 = 0.8 = 0 + 0.8$$

$$0.8 * 2 = 1.6 = 1 + 0.6$$

$$0.6 * 2 = 1.2 = 1 + 0.2$$