

Задание 1.2

Задача 1.2.1

- a) Рассматриваем диапазон чисел от 0 до 63 включительно.
- b) Выполняем деление числа на 2, фиксируя остаток от деления.
- c) Продолжаем делить на 2, пока результат не станет равным 0, и записываем остатки в обратном порядке.
- d) Добавляем слева недостающие нули, чтобы итоговое представление имело 6 бит.

000000.

- e) 13

001101.

- f) 24

011000.

- g) 63

011000.

Задача 1.2.2

Проверка диапазона, от -32 до 31 включительно. Преобразовать в двоичное представление, если число равно нулю или положительное то оставляем как было, если отрицательное то инвертируем и прибавляем 1

- a) 16

011000.

- b) -2

011000.

- c) 31

011000.

- d) -32

011000.

Задача 1.2.3

Проверка длины, преобразование (Каждую цифру двоичного числа умножаем на 2^n , где n – позиция бита, начиная с 0 справа), суммирование (сложение всех полученных значений), знак (если старший бит равен 1, число отрицательное)

- a) 000101

5

b) 101011

41

23

c) 111111

1

63

d) 100000

32

-32

Задача 1.2.4

Проверка диапазона от 0 до 255 включительно, преобразование в двоичное представление, дополнение до 8 бит, преобразование в шестнадцатеричное представление

a) 0

1. 00000000

2. 00

b) 10

1. 00001010

2. 0A

c) 14

1. 00001110

2. 0E

d) 15

1. 00001111

2. 0F

e) 16

1. 00010000

2. 10

f) 32

1. 00100000

2. 20

g) 34

1. 00100010

2. 22

h) 127

1. 01111111

2. 7F

i) 128

1. 10000000

- 2. 80
- j) 255
- 1. 11111111
- 2. FF

Задача 1.2.5

Проверка диапазона, преобразование в двоичное представление, один F это 4 бита, дополнить до 8 бит

- a) 0x1: 00000001
- b) 0x2: 00000010
- c) 0x7: 00000111
- d) 0x8: 00001000
- e) 0x10: 00010000
- f) 0x7F: 01111111
- g) 0xFF: 11111111
- h) 0x80: 10000000

Задача 1.2.8

- a) Big-endian: CA FE FE ED
- b) Little-endian: ED FE FE CA