



1. ans

- i. 110
- ii. 100011110
- iii. 10000000000
- iv. 0.01
- v. 111.001
- vi. 10.101

2. ans

- i. 55
- ii. 157
- iii. 0.625
- iv. 2.25
- v. 10.125

3. ans

- i. 8进制
  - a. 233.154
  - b. 1252.144
- ii. 16进制
  - a. 9b.36
  - b. 2aa.32

4. ans

- i. 111101.11000101
- ii. 11001001010.110000111111

5. ans

- i. 解释如下

## 1. 原码 (Sign-Magnitude)

**原码** (也称为符号-数值表示) 是最简单的二进制表示方法, 用来表示正负整数。

- **表示方式**: 最高位 (符号位) 表示正负号, 剩下的位表示该数值的绝对值。
  - **符号位**: 0 表示正数, 1 表示负数。
  - **数值位**: 表示数字的大小。

例如:

- 对于 8 位二进制:

- +5 的原码: 0000 0101
- -5 的原码: 1000 0101

**优点:** 简单直观, 易于理解。

**缺点:** 存在两个零 (+0 和 -0), 计算复杂 (如加减法需要考虑符号位处理)。

## 2. 反码 (Ones' Complement)

**反码** 是一种将负数表示为正数的逐位取反的编码方式。

- **正数:** 反码与原码相同。
- **负数:** 反码通过将对应正数的每一位取反 (0 变成 1, 1 变成 0) 得到。

例如:

- 对于 8 位二进制:
  - +5 的反码: 0000 0101 (与原码相同)
  - -5 的反码: 1111 1010 (逐位取反 +5)

**优点:** 反码较原码更有助于进行二进制运算, 尤其是减法。

**缺点:** 仍然存在两个零 (+0 和 -0), 并且计算时需要处理进位。

## 3. 补码 (Two's Complement)

**补码** 是现代计算机中最常用的有符号数表示方法, 它通过解决反码中的双零问题来简化计算。

- **正数:** 补码与原码相同。
- **负数:** 补码是对应正数的反码加 1。

例如:

- 对于 8 位二进制:
  - +5 的补码: 0000 0101 (与原码相同)
  - -5 的补码: 1111 1011 (反码 1111 1010 再加 1)

**优点:**

- **只有一个零:** 补码表示系统中只有一个零 (0000 0000)。
- **运算简单:** 补码系统中, 加减法运算与无符号数的运算相同, 无需额外处理符号位。

**缺点:** 负数的补码表达不如原码直观, 但因为计算机中不需要直观理解, 补码仍然是最优选择。

# 总结

- **原码** 直接表示符号和数值，简单但存在双零问题。
- **反码** 通过取反表示负数，计算上有一定改进，但仍然有双零问题。
- **补码** 通过加 1 解决了反码的问题，计算简单，是现代计算机中广泛使用的数值表示方法。

通过补码表示方式，计算机能够有效进行二进制加减法操作，并处理正负数转换的细节。

ii. ans  
1.

Number	Sign-Magnitude	Ones' Complement	Two's Complement
127	01111111	01111111	01111111
-127	11111111	10000000	10000001
135	10000111	10000111	10000111
-120	11111000	10000111	10001000

Number	IEEE 754 Binary Representation
0.75	00111111010000000000000000000000
-0.75	10111111010000000000000000000000

6. ans:0.125-15  
7. ans

## 1. 位 (bit)

位是计算机数据的最小单位。它只有两种可能的状态，通常表示为 0 或 1。在二进制系统中，位用来表示数据的基本信息。

## 2. 字节 (byte)

字节是由 8 位组成的一个数据单元。在大多数计算机系统中，1 个字节可以表示 256 个不同的值 ( $2^8 = 256$ )，它是数据存储的基本单位。例如，一个字符（如字

母 A) 通常占用 1 个字节的空間。

### **3. 字 (word)**

字是计算机进行处理时数据的基本单元。一个字的长度可以根据计算机的体系结构不同而变化。通常，字的长度是 16 位、32 位或 64 位。字的长度决定了处理器一次能处理的数据量。

### **4. 字长 (word length)**

字长是指计算机内部数据处理单元的宽度，也就是处理器一次能够处理的最大二进制位数。例如，对于 32 位系统，字长为 32 位；对于 64 位系统，字长为 64 位。字长影响计算机的处理能力和存储地址空间的大小。