

1.

**Q: 浮点数有没有移位操作和扩展操作?为什么?**

**A:** 浮点数在计算机中是以一定的格式存储的, 通常包括符号位、指数位和尾数位。这种存储方式使得浮点数的计算更加高效和精确。在这种表示下, 浮点数的移位和扩展操作不会直接应用于其存储格式, 因为这可能会破坏浮点数的结构, 导致结果不符合浮点数的规范。

2.

**Q: 为什么整数除 0 会发生异常而浮点数除 0 不会?**

**A:** 整数除以 0 会引发异常, 因为在数学上除数不能为 0, 这会导致数学上的无解情况, 因此大多数编程语言会在整数除以 0 时抛出异常或错误。

而在浮点数运算中, 当浮点数除以 0 时, 通常会得到无穷大 (Infinity) 或非数值 (NaN) 的结果, 而不会抛出异常。这是因为浮点数的运算规则在处理分母为 0 的情况时做了特殊处理, 以确保计算过程的平稳进行。当分母为 0 时, 计算结果被定义为无穷大或无穷小, 以便在后续计算中保持数值稳定性。

3.

**Q: 结合例子说明为何 IEEE 754 加减运算右规时最多只需一次?**

**A:** 因为即使是两个最大的尾数相加, 得到的和的尾数也不会达到 4, 故尾数的整数部分最多有两位, 保留一个隐含的“1”后, 最多只有一位被右移到小数部分。

4.

**Q: 为什么同一个实数赋给 float 型变量和 double 型变量, 输出结果会有所不同?**

**A:** float 和 double 的精度不同。float 只有 32 位, 而 double 有 64 位, 因此 double 可以表示更大范围的数字和更高的精度。当同一个实数赋给 float 和 double 时, float 会将数字四舍五入到最接近的 32 位二进制数, 而 double 会将数字四舍五入到最接近的 64 位二进制数, 因此结果不同。