1. 이진법으로 무한한 수를 처리할 수 있다?

이진법으로 모든 양의 정수를 나타낼 수 있기 때문에 무한한 양의 정수는 처리할 수 있다. 하지만, 음의 정수를 처리할 경우 부호를 나타내기 위한 별도의 장치 또는 약속이 필요하며 이는 소수 또한 마찬가지이다.

2. 컴퓨터로 무한한 수를 처리할 수 있다?

컴퓨터는 계산기이다. 계산을 하기 위해서는 계산할 값들을 저장할 공간과 계산 결과를 저장할 공간이 필요하다. 메모리 등 물리적인 공간에 전기적으로 그러한 값들을 저장하는데 저장 공간의 물리적인 한계가 존재할 수 밖에 없다. 때문에 컴퓨터로 무한한 수를 처리할 수 없다.

3. 두 양수의 덧셈 결과는 항상 양수이다?

일반적인 두 양수의 덧셈 결과는 항상 양수이지만 본 소스 코드의 결과에서 양의 정수인 1234567890과 1234567890을 더했을 때 음수값이 도출되었다. 이는 Int 자료형을 저장하는 메모리 공간의 크기가 macOS 기준 32바이트인데 결과값이 이 범위를 넘어서 오버플로우가 발생해 음수로 해석되었기 때문이다. 그냥 12345678901234567890을 %d로 int 자료형으로 표현했을 때도 같은 이유이다.

4. 부동 소수점 연산으로 정확한 소수 연산을 할 수 있다?

1.1을 %.20f로 소수점 아래 20자리까지 1.1000000000000000008882으로 출력했는데 이는 컴퓨터가 사용하는 부동 소수점 연산의 한계라고 할 수 있다. 이진법으로 소수점 아래 자리수를 나타내기 때문에 메모리 공간의 크기 또는 int 자료형의 크기로 인해 나타낼 수 있는 소수점 아래 표현의 한계로 인해 이러한 문제가 발생한다. 0.125, 0.25, 0.5 등 범위 내에서 이진법으로 표현 가능한 소수만 오차 없이 정확한 소수 연산을 할 수 있다.