

과제

전장훈

아래와 같이 출력되는 이유를 알아보자.

```
print(10 / 3)
3.3333333333333335
```

```
0.1 + 1.1 == 1.2
False
```

답 : 컴퓨터가 실수를 저장할 때, 지수부와 가수부로 나누어 저장하게 되기 때문이다.

부연 설명

컴퓨터가 실수를 저장할 때 부동소수점 표현으로 저장하게 된다. 부동소수점 표현에서 숫자는 부호부, 지수부, 가수부의 세 부분으로 구성된다.

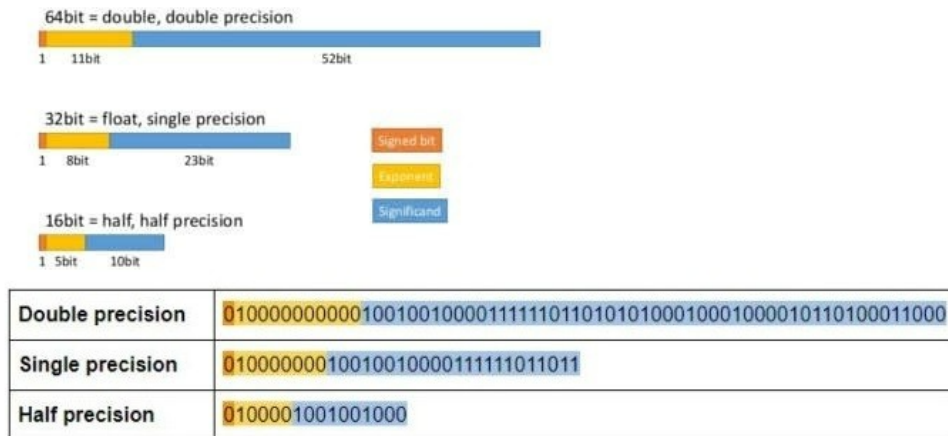
$$N = (-1)^S \times M \times 2^E$$

- 부호부 (Sign) : 1비트. 숫자의 부호를 나타내며, 양수일 때 0, 음수일 때 1이다.
- 지수부 (Exponent) : 8비트. 지수를 나타낸다.
- 가수부 (Mantissa) : 23비트. 가수 또는 유효숫자를 나타낸다.

부동 소수점 산술에 대한 IEEE 표준은 컴퓨터에서 이진수로 숫자를 나타내는 일반적인 규칙이다. 배정 밀도 형식에서 각 숫자는 64 비트를 차지하고, 단정밀도 형식은 32 비트를 사용하지만 반정밀도는 16 비트이다. 그러나 컴퓨터는 이 정보를 부동 소수점, 숫자 및 해당 지수를 나타내는 일련의 1과 0 (이 경우 1.1001001×2^1)으로 이진으로 저장한다.

단정도 32 비트 형식에서는 숫자가 양수인지 음수인지를 나타내는 데 1 비트가 사용된다. 8 비트는 지수를 위해 예약되어 있는데, 이진수이기 때문에 2의 거듭 제곱이 된다. 나머지 23 비트는 부호로 불리는 숫자를 구성하는 숫자를 나타내는 데 사용된다. 배정 밀도는 지수의 경우 11 비트, 유효의 경우 52 비트를 예약하여 표시 할 수 있는 숫자의 범위와 크기를 크게 확장한다.

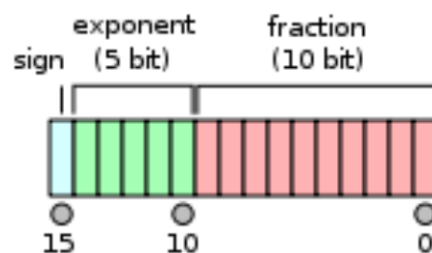
Format of Floating points IEEE754



0.625를 이진수로 변환하면,

- $0.625 * 2 = 1.25 \rightarrow$ 정수부분 1
- $0.25 * 2 = 0.5 \rightarrow$ 정수부분 0
- $0.5 * 2 = 1.0 \rightarrow$ 정수부분 1
- $0.0 * 2 = 0$ 종료
- 결과: 0.101

컴퓨터의 고정 소수점 레지스터 방식



그런데 위 식에서 봤다시피 0.1은 0001100110011... 이런식으로 계속 반복된다. 할당된 10자리를 벗어나게 된다. 저 방식을 무시하고 계속해서 저장하게 되면 컴퓨터 메모리는 0.1의 값을 저장하는데 할당되기 때문에, 저런 저장 방식을 따라야만 한다.

결국 저런 방식으로 인해, 정확한 값이 아닌 근삿값만 얻을 수 있게 된다. 이 값들을 더하면 당연히 정확한 값을 기대하기는 어렵다.