IEEE Standard Numbers

For a double (single) precision number, the sign $s = \pm 1$ takes 1 bit, the exponent has 11 bits (8), and the mantissa $0.1\, f_1\, f_2 \ldots$ has 52 bits (23).

In reading the following bear in mind that

$$0.1\, f_1\, f_2\, \ldots \times 2^E = 1.f \times 2^{E-1}\,.$$

The exponent is stored in biased form. This means that the exponent $E = e - 1024$ where $0 < e < 2047$. This trick means that the exponent is stored as a positive integer e less than 2047.

The leading 1 for a non zero mantissa is not stored. This means that since the leading digit of the mantissa of a nonzero number always is equal to one, there is no need to store it.

The complete definition for a double precision (single precision) number is the following. The meat of the definition is part (c). The others just specify how that largest and smallest numbers are handled.

(a) If $e = 2047$ (255) and $f \neq 0$, then NaN, i.e., not a number.

(b) If $e = 2047$ (255) and $f = 0$, then $-1^s \times \infty$.

(c) If $0 < e < 2047$ (255), then $(-1^s)(1.f)(2^{e-1023})$ (or $(-1^s)(1.f)(2^{e-127})$).

(d) If $e = 0$ and $f \neq 0$, then $(-1^s)(0.f)(2^{-1022})$ (or $(-1^s)(0.f)(2^{-126})$).

(e) If $e = 0$ and $f = 0$, then $-1^s \times 0$.

Though it is not needed for this problem it is useful to know what the IEEE standard is for rounding a number which is exactly half way between the two nearest floating point numbers. The recipe is that one considers the 53 bit mantissas of the nearest neighbors. Both start with one and are followed by 52 digits each a 1 or 0. It is not hard to see that one of the neighboring mantissas is even, that is ends with a zero, and the other is odd. The IEEE standard is to round to the even one. This arranges that the rounding is half the time up and half the time down in a random way.