How RSA Works (generalized form)

The RSA system works with numbers that were originally strings of letters turned via ASCII into these numbers. The numbers are encoded by application of a mathematical function f, and then decoded using the function's inverse f^{-1}. The last step is to turn the numbers back into strings of letters.

RSA Procedure:

1. Pick two primes p, q
2. Set $n = pq$ and calculate $\phi(n) = (p-1)(q-1)$
3. Pick an odd number a such that $1 < a < \phi(n)$ and $\gcd(\phi(n), a) = 1$
4. Compute b, the multiplicative inverse of a mod $\phi(n)$ [i.e., such that ab is congruent to 1 (mod $\phi(n)$)].
5. Publish (a, n) as the public key. Retain b as the private key.

Encoding Message M: send $C = M^a \pmod{n}$ [i,e, f(M)]

Decoding Message C: compute $M = C^b \pmod{n}$ [i.e. $f^{-1}(C)$]

Note: In step 4, we use the power of Maple (via the function "inverse of a mod m", not the fraction 1/a) to calculate b directly with the line:

b:= 1/a mod ϕ(n);

This gives b immediately.

Note: The RSA encryption works because:

$$
\begin{aligned}
C^b \pmod n &= \left(M^a \pmod n\right)^b \pmod n && [\textit{apply Law of Mod Mult}] \\
&= \left(M^a\right)^b \pmod n = \left(M^{ab}\right)\pmod n \\
&= \left(M^{1+t(p-1)(q-1)}\right)\pmod n && [\textit{ for some } t] \\
&= (M)\left(M^{t(p-1)(q-1)}\right)\pmod n && [\textit{apply Euler}; n = pq] \\
&= M \pmod n = M && [\textit{because } M < n]
\end{aligned}
$$