make a 2-column list of all pairs n, g^n as n goes from 1 to $q-1$; then make third and fourth columns listing all pairs a, $log_g a$. That is, list the elements a of $\mathbf{F}_q^*$ in some convenient order in the third column, and then run down the first two columns, putting each n in the fourth column next to the a which is g^n. For example, to do this for $\mathbf{F}_9$ (see Example 2 in § II.1), we choose $g = \alpha$ to be a root of $X^2 - X - 1$, and make the following table:

n	g^n	a	$log_g a$
1	α	1	8
2	$\alpha+1$	-1	4
3	$-\alpha+1$	α	1
4	-1	$\alpha+1$	2
5	$-\alpha$	$\alpha-1$	7
6	$-\alpha-1$	$-\alpha$	5
7	$\alpha-1$	$-\alpha+1$	3
8	1	$-\alpha-1$	6

Then multiplication or division involves nothing more than addition or subtraction modulo $q-1$ and looking at the table. For example, to multiply $\alpha-1$ by $-\alpha-1$, we find the two numbers in the third column, add the two corresponding logarithms: $7+6 \equiv 5 \ mod\ 8$, and then find the answer $-\alpha$ in the second column next to 5.

(a) Make a log table for $\mathbf{F}_{31}^*$, and use it to compute $16 \cdot 17$, $19 \cdot 13$, $1/17$, $20/23$.

(b) Make a log table for $\mathbf{F}_8^*$, and use it to compute the following (where α is a root of X^3+X+1; your answers should not involve any higher power of α than α^2): $(\alpha+1)(\alpha^2+\alpha)$, $(\alpha^2+\alpha+1)(\alpha^2+1)$, $1/(\alpha^2+1)$, $\alpha/(\alpha^2+\alpha+1)$.

2. At first glance, it may seem that we could use the cyclic group $(\mathbf{Z}/p^\alpha\mathbf{Z})^*$ (see Exercise 2(a) in § II.1) instead of $\mathbf{F}_q^*$ as a setting for the discrete logarithm problem. However, the discrete log problem for $(\mathbf{Z}/p^\alpha\mathbf{Z})$ for $\alpha > 1$ turns out to be essentially no more time-consuming (even if α is fairly large) than for $\alpha = 1$ (i.e., $\mathbf{F}_p$). More precisely, using the same technique that is given below in this exercise, one can prove that, once one solves the discrete log problem modulo p, going the rest of the way (i.e., solving it modulo p^α) takes polynomial time in $log(p^\alpha) = \alpha \log p$. (Recall that no algorithm is known which solves the discrete log problem modulo p for large p in polynomial time in $\log p$; and experts doubt that such an algorithm exists.) In this exercise, we show that in the case $p = 3$ there's a straightforward algorithm which solves the discrete log problem modulo 3^α in time which is polynomial in α.

Thus, suppose we take $g = 2$ (it is easy to show that 2 is a generator of $(\mathbf{Z}/3^\alpha\mathbf{Z})^*$ for any α), we have some integer a not divisible by 3, and we want to solve the congruence $2^x \equiv a \ mod\ 3^\alpha$. Prove that the following