Homework 22, due October 26

(1) Recall that in the Pollard rho method of factoring, we choose a polynomial $f(u)$ and a seed u_0. For $i \geq 1$, we define u_i recursively as $u_i = f(u_{i-1})$. The idea is that if $n = pq$, the u_i will start repeating $\pmod p$ before they repeat $\pmod n$, so we should have $u_i \equiv u_j \pmod p$ for some i, j. We don't want to check $\gcd(u_i - u_j, n)$ for every i, j, so instead we check $\gcd(u_{2s} - u_s, n)$ for $s = 1, 2, 3, \ldots$ since eventually we'll find an s that's a multiple of the cycle length.

In SAGE code it might look like this: (Here $f(u) = u^2 + 1$ and $n = 36287$.)

```
u=[1]

for i in range(1,20):
    u.append(lift(mod(u[i-1]^2+1, 36287)))

for s in range(1,10):
    print(gcd(u[2*s]-u[s], 36287))
```

Factor $16019, 10277$, and 199934971 using the Pollard rho method. How large does s get before you find a factor? Look at the numbers $u_i \pmod p$, where p is the factor you found. How long is the cycle in each case?

(2) (Page 198, problem 5) Factor

`8834884587090814646372459890377418962766907`

by the $p - 1$ method.

(3) The ciphertext

`570360711957965038148054313442031747824957123638823375528569417305522`

was encrypted with RSA with public key (n, e) given by

`(1849984765134873910404765458412903449879887030956920096187415338501539, 9007).`

The prime factors p and q of n are consecutive primes. Decrypt.