Fast Modular Exponentiation

Fermat's / Euler's $\Rightarrow$ $a^{power} \bmod n$

Need to be able to raise a number to a high exponent mod another number!

Standard

```
ans = 1
for (int i = 0; i < exp; i++)
    ans = (ans * base) % mod;
```

Run time $= O(exp \cdot (amt\ time\ mult + mod))$

↑ usually d^2, d = # digits in mod value

↳ NOT POSSIBLE is exp has say 200 digits

SPEED UP

$$base^{exp} = \left(base^{exp/2}\right)\left(base^{exp/2}\right)$$ if exp is even!

$47^{100} = 47^{50} \times 47^{50}$

Under mod $\Rightarrow$ $(47^{50} \bmod n)^2$

Save then square it

Save 50 multiplications!

```
int fastmodexpo(int base, int exp, int mod) {
    if (mod == 1) return 0;
    if (exp == 0) return 1;
    if (exp % 2 == 0) {
        int tmp = fastmodexpo(base, exp/2, mod);
        return (tmp*tmp) % mod;
    }
    return (base*fastmodexpo(base, exp-1, mod)) %
                                            mod;
    // b^e = b * b^(e-1)
```

Runtime = $O((\log exp) * (\text{time mult/mod}))$

QUIZ 4 - NUM THEORY

Prime Factorization
Formula ϕ (euler phi function)
Fermet's Thm
Euler's Thm
Fermet Factoring
Miller Rabin Primality Test
Discrete Log Problem (efficiently work out ~~exp each~~ a base raised to each exponent mod some value)
Primitive Roots/Generators (how many + how to obtain)
Fast Mod Expo

Calculator (make sure know how to do mod)
No Notes

$3^9 \bmod 13$ for each 9

exp	0	1	2	3	4	5	6	7	8	9	10	11	12
$3^{exp} \bmod 13$	1	3	9	1	3	9	1	3	9	1	3	9	1
2	1	2	4	8	3	6	12	11	9	5	10	7	1

$9 * 3 = 27 \bmod 13 = 1$

$8 \cdot 2 = 16 \equiv 3 \bmod 13$
$12 \cdot 2 = 24 \equiv 11 \bmod 13$
$11 \cdot 2 = 22 \equiv 9 \bmod 13$
$9 \cdot 2 = 18 \equiv 5 \bmod 13$
$10 \cdot 2 = 20 \equiv 7 \bmod 13$
$7 \cdot 2 = 14 \equiv 1 \bmod 13$