Asgn 6

*For mpz, must init

to get username .pub

*look at gmp man page

1) gotta choose 2 large prime #s,
2) random int $2 < e < n \ni gcd(c, \varphi(n))=1$

#making 3 executabues

randstate.c

```
void randstate_init (uint64_t seed){
        gmp_randinit_mt(num);
        gmp_randseed_ui(num, seed);
}
```

```
void randstate_clear(){
        gmp_randclear(num);
}
```

numtheory.c   #look @ gmp

```
void pow_mod(mpz_t out, mpz_t base, mpz_t exponent, mpz_t mod){
        mpz_set_ui(mpz_t out, 1);
        mpz_init(p);
        mpz_init(zero);
        mpz_set(mpz_t p, mpz_t base);
        while (mpz_cmp(mpz_t exponent, mpz_t zero){
              if (!(mpz_t % 2)){
                    mpz_mul(mpz_t out, mpz_t out, mpz_t p)
                    mpz_t out mod modulus;
              mpz_mul(mpz_t p , mpz_t p , mpz_t p)
              mpz_t out mod modulus
              mpz_cdiv_q_ui(mpz_t d, mpz_t d, 2);  #do floor
    return mpz_t out;
```

```
void is_prime(mpz_t n, uint64 iters){
# write n-1 = 2^s r such that r is odd
for (uint64_t i = 1; i >= iters; i++){
        # for the random what do i use #what is r?
         mpz_init (x)
        mpz_sub_ui(x, mpz_t n, 1)
       if (y != 1 && y != (x){
```

Keygen

p-makeprime

pub $\left\{ \begin{array}{l} n=p\cdot q \\ e= 65537 \end{array} \right.$

$\underset{\uparrow}{d}= \frac{1}{65537} \% (p-1)(q-1)$

prive

encrypt

·pub

1) message (large bits) (read ascii) = m

$m^e \% n$ = encryted E

dycript

·prive

$E^d \% n$ = original file

Mod-inverse

-make compies (temp) for r, r' to avoid issues