Sử dụng thư viện GMP

Header và thư viện

```
#include <gmp.h>
```

Biên dịch với `gcc`

```
gcc myprogram.c -lgmp
```

Sinh khoá RSA dùng thư viện GMP

- Input: p, q là hai số nguyên tố; và e là số mũ công khai (có thể bằng 3 hoặc 65537)
- Output:

modun	Khoá công khai	Khoá bí mật
$n = p \cdot q$	(n, e)	d

Thuật toán:

1. Tính $n = p \cdot q$
2. Tính $\phi(n) = (p - 1) \cdot (q - 1)$
3. Tính $d = e^{-1} \mod \phi(n)$

```
void rsa_keys(mpz_t n, mpz_t d, const mpz_t p, const mpz_t q, const mpz_t e) {
    mpz_mul(n, p, q);

    mpz_t p_1, q_1, phi;
    mpz_init(p_1);  mpz_init(q_1); mpz_init(phi);

    mpz_sub_ui(p_1, p, 1);
    mpz_sub_ui(q_1, q, 1);
    mpz_mul(phi, p_1, q_1);          // tính phi(n)

    mpz_t gcd;
    mpz_init(gcd);
    mpz_gcd(gcd, e, phi);
    assert(mpz_cmp_ui(gcd, 1) == 0);  // gcd (e, phi) == 1
                                      // mới có nghịch đảo
    mpz_invert(d, e, phi);            // d = e^-1 mod phi(n)

    mpz_clear(gcd);
    mpz_clear(p_1);
    mpz_clear(q_1);
}
```

Bài tập

Hãy viết hàm mã hoá và giải mã như trong Textbook RSA

```
void encrypt(mpz_t c,
             const mpz_t m,
             const mpz_t e,
             const mpz_t n) {
        // TODO
}
void decrypt(mpz_t m,
             const mpz_t c,
             const mpz_t d,
             const mpz_t n) {
        // TODO
}
```

Một số hàm tính toán liên quan đến số học modun:

- `void mpz_gcd (mpz_t g, const mpz_t a, const mpz_t b)` : $g = \gcd(a, b)$
- `void mpz_gcdext (mpz_t g, mpz_t x, mpz_t y, const mpz_t a, const mpz_t b)` :

$(g, x, y) = \text{gcdx}(a, b)$

- `void mpz_powm (mpz_t z, const mpz_t a, const mpz_t b, const mpz_t n)`:
  $z = a^b \mod n$