กลุ่ม : G01 ข้อที่ 1

6509611809 ธีรภัทร ศิริธรรม

6509611544 กิตติธรา สุทธาภิรมย์

6509611858 ประพล ขาวสอาด

Code ในข้อที่ 1

```c
#include <stdio.h>
#include <openssl/bn.h>

void printBN(char *msg, BIGNUM * a)
{
/* Use BN_bn2hex(a) for hex string
* Use BN_bn2dec(a) for decimal string */
char * number_str = BN_bn2hex(a);
printf("%s %s\n", msg, number_str);
OPENSSL_free(number_str);
}


int main(){
    const char *P_STR = "F7E75FDC469067FFDC4E847C51F452DF";
    const char *Q_STR = "E85CED54AF57E53E092113E62F436F4F";
    const char *E_STR = "0D88C3";

    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *p = BN_new();
    BIGNUM *q = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *phi_n = BN_new();
    BIGNUM *one = BN_new();
    BIGNUM *pkey = BN_new();
    BN_hex2bn(&one , "1");

    BN_hex2bn(&p , P_STR);
    BN_hex2bn(&q , Q_STR);
    BN_hex2bn(&e , E_STR);

    BN_sub(p ,p , one);
    BN_sub(q ,q , one);
    BN_mul(phi_n , p , q , ctx);

    BN_mod_inverse(pkey , e , phi_n , ctx);
    printBN("Private Keys :" , pkey);
```

```
    BN_CTX_free(ctx);
    BN_free(p);
    BN_free(q);
    BN_free(phi_n);
    BN_free(e);
    BN_free(one);
    BN_free(pkey);

}
```

อธิบายส่วนที่ 0 : ติดตั้ง เตรียมพร้อมสภาพแวดล้อม include openssl library สำหรับจัดการตัวเลขขนาดใหญ่

```
#include <stdio.h>
#include <openssl/bn.h>
```

อธิบายส่วนที่ 1 : p , q , e เป็น 3 prime number โดยในที่นี้เราจะใช้ e , n มาเป็น public key

```
    const char *P_STR = "F7E75FDC469067FFDC4E847C51F452DF";
    const char *Q_STR = "E85CED54AF57E53E092113E62F436F4F";
    const char *E_STR = "0D88C3";
```

อธิบายส่วนที่ 2 : convert char* to BIGNUM (Type ตัวเลขสำหรับ library openssl) เพื่อเตรียมข้อมูลให้พร้อมสำหรับการคำนวณ

```
    BN_CTX *ctx = BN_CTX_new();
    BIGNUM *p = BN_new();
    BIGNUM *q = BN_new();
    BIGNUM *e = BN_new();
    BIGNUM *phi_n = BN_new();
    BIGNUM *one = BN_new();
    BIGNUM *pkey = BN_new();
    BN_hex2bn(&one , "1");

    BN_hex2bn(&p , P_STR);
    BN_hex2bn(&q , Q_STR);
    BN_hex2bn(&e , E_STR);
```

อธิบายส่วนที่ 3 : ลดค่า p , q ไป 1 จากนั้นเราจำคำนวณหา phi n จาก p x q (ลงลง 1 แล้ว)

หมายเหตุ : เนื่องจาก phi(n) = (p-1)(q-1) และ p , q เป็น prime number

```
    BN_sub(p ,p , one);
    BN_sub(q ,q , one);
    BN_mul(phi_n , p , q , ctx);
```

อธิบายส่วนที่ 4 : จากนั้นเราจะคำนวณโดยใช้ฟังก์ชั่น BN_mod_inverse ซึ่งจะคำนวณสมการ

$$d = e^{-1} mod\ \varphi(n)$$

pkey = private key (d)

e = ค่า e ของ public key

phi_n = ค่า phi ของ n ซึ่งคำนวณได้มาจากขั้นตอนที่แล้ว

ctx คือ context structure for BIGNUM operations สำหรับ library OpenSSL

เมื่อคำนวณเสร็จเราจำทการพิมพ์ค่า private key ออกมา

```
BN_mod_inverse(pkey , e , phi_n , ctx);
printBN("Private Keys :" , pkey);
```

อธิบายส่วนที่ 5 : free memory ให้เรียบร้อยก่อยจบโปรแกรม

```
BN_CTX_free(ctx);
    BN_free(p);
    BN_free(q);
    BN_free(phi_n);
    BN_free(e);
    BN_free(one);
    BN_free(pkey);
```