#include <stdio.h>

#include <math.h>

long long power(long long a, long long b, long long p) {

long long res = 1;

a = a % p;

while (b > 0) {

if (b % 2 == 1)

res = (res \* a) % p;

b = b >> 1;

a = (a \* a) % p;

}

return res;

}

long long modInverse(long long a, long long p) {

return power(a, p - 2, p);

}

int main() {

long long p = 23;

long long g = 5; y

long long y = power(g, x, p);

printf("Public Key: (p=%lld, g=%lld, y=%lld)\n", p, g, y);

printf("Private Key: x = %lld\n", x);

long long M = 15;

long long k = 10;

long long a = power(g, k, p);

long long b = (M \* power(y, k, p)) % p;

printf("Encrypted: a = %lld, b = %lld\n", a, b);

long long s = power(a, x, p);

long long M\_decrypted = (b \* modInverse(s, p)) % p;

printf("Decrypted Message: %lld\n", M\_decrypted);

return 0;

}

