#include <stdio.h>

// Function to compute `a^m mod n`

int compute(int a, int m, int n)

{

int r;

int y = 1;

while (m > 0)

{

r = m % 2;

// fast exponention

if (r == 1) {

y = (y\*a) % n;

}

a = a\*a % n;

m = m / 2;

}

return y;

}

// C program to demonstrate the Diffie-Hellman algorithm

int main()

{

int p = 23; // modulus

int g = 5; // base

int a, b; // `a` – Alice's secret key, `b` – Bob's secret key.

int A, B; // `A` – Alice's public key, `B` – Bob's public key

// choose a secret integer for Alice's private key (only known to Alice)

a = 6; // or, use `rand()`

// Calculate Alice's public key (Alice will send `A` to Bob)

A = compute(g, a, p);

// choose a secret integer for Bob's private key (only known to Bob)

b = 15; // or, use `rand()`

// Calculate Bob's public key (Bob will send `B` to Alice)

B = compute(g, b, p);

int keyA = compute(B, a, p);

int keyB = compute(A, b, p);

printf("Alice's secret key is %d\nBob's secret key is %d", keyA, keyB);

return 0;

}