**/\* Operaciones con Matices \*/**

#include <bits/stdc++.h>

**using** **namespace** **std**;

**struct** matrix {

**int** mat[5][5], **int** N;

matrix(int a=5) {

N = a;

**for**(**int** i=0; i<N; i++)

**for**(**int** j=0; j<N; j++)

mat[i][j] = 0;

}

**void** **operator**=(matrix A) {

**for**(**int** i=0; i<N; i++)

**for**(**int** j=0; j<N; j++)

mat[i][j] = A.mat[i][j];

}

};

// Suma de matrices

// Calcula C = A + B en O(N^2).

matrix sum(matrix A, matrix B) {

matrix C(A.N);

**int** N = A.N;

for(**int** i=0; i<N; i ++)

for(**int** j=0; j<N; j++)

C.mat[i][j] = A.mat[i][j] + B.mat[i][j];

**return** C;

}

// Producto de matrices

// Calcula C = A\*B en O(N^3).

matrix dot(matrix A, matrix B) {

matrix C(A.N);

**int** N = A.N;

**for**(**int** i=0; i<N; i ++)

**for**(**int** j=0; j<N; j++)

**for**(**int** k=0; k<N; k++)

C.mat[i][j] += A.mat[i][k] \* B.mat[k][j];

**return** C;

}

// Potencia de matrices

// Calcula B = A^k en O(N^3\*log(k)).

matrix pow(matrix A, int k) {

**if**(k == 1)

**return** A;

matrix C(A.N);

C = pow(A, k/2);

**if**(k%2==0) **return** dot(C, C);

**else** **return** dot(dot(C, C ),A);

}