Bài yêu cầu in ra luỹ thuỹ của 1 ma trận (sử dụng luỹ thừa nhị phân)

Trong thuật toán luỹ thừa nhị phân cần phải có các bước

Để nhân 2 ma trận ta sử dụng vector 2 chiều để lưu và trả về ma trận dễ dàng

```
long long exp(long long a,long long b){
   if(b==1) return a;
   if(b==0) return 1;
   if(b%2) return ((exp(a,b/2)%mod *exp(a,b/2)%mod)%mod*a)%mod ;
   else return (exp(a,b/2)%mod *exp(a,b/2)%mod)%mod;
}
```

+ nhân 2 số → nhân 2 ma trận

→ chỉ cần xây dựng 1 hàm nhân 2 ma trận và trả về ma trận kết quả dưới dạng vector

Code:

```
#include<iostream>
#include<vector>
using namespace std;
typedef vector<vector<long long>> vll;
const int mod=1e9+7;
int n,step;
vll multiply(vll A,vll B) {
   vll C(n);
   for (int i = 0 ; i < n ; i++) {
      for (int j = 0 ; j < n ; j++) {
         long long sum = 0;
         for (int k = 0 ; k < n ; k++) {
            sum += A[i][k] * B[k][j];
            sum%=mod;
         }
         C[i].push_back(sum);
      }
   }
   return C;
}

vll binary_exponentiation(vll A ,int exp) {
   if (exp == 1) return A;
   vll temp = binary_exponentiation(A,exp/2);
   if(exp&1) return multiply(multiply(A,temp),temp);
   else return multiply(temp,temp);
}
```

```
int main() {
    cin >> n >> step;
    vll A(n);
    for (int i = 0 ; i < n ; i++) {
        for (int j = 0 ; j < n ; j++) {
            int x; cin >> x;
            A[i].push_back(x);
        }
    }
    vll result = binary_exponentiation(A,step);
    for(int i = 0 ;i < n ; i++){
        for(auto x: result[i])
            cout<<x<<" ";
        cout<<endl;
    }
}
```