#include <iostream>

#include <vector>

using namespace std;

#define ll long long

#define MOD 1000000000

ll k;

vector<ll> a,b,c;

//Multiply two matrices

vector<vector<ll> > multiply(vector<vector<ll> > A,vector<vector<ll> > B ){

//third matrix mei result store

vector<vector<ll> > C(k+1,vector<ll>(k+1));

for(int i=1;i<=k;i++){

for(int j=1;j<=k;j++){

for(int x=1;x<=k;x++){

C[i][j] = (C[i][j] + (A[i][x]\*B[x][j])%MOD)%MOD;

}

}

}

return C;

}

vector<vector<ll> > pow(vector<vector<ll> > A,ll p){

//Base case

if(p==1){

return A;

}

//Rec Case

if(p&1){

return multiply(A, pow(A,p-1));

}

else{

vector<vector<ll> > X = pow(A,p/2);

return multiply(X,X);

}

}

ll compute(ll n){

//Base case

if(n==0){

return 0;

}

//Suppose n<=k

if(n<=k){

return b[n-1];

}

//Otherwise we use matrix exponentiation, indexing from 1

vector<ll> F1(k+1);

for(int i=1;i<=k;i++){

F1[i] = b[i-1];

}

//2. Transformation matrix

vector<vector<ll> > T(k+1,vector<ll>(k+1));

// Let init T

for(int i=1;i<=k;i++){

for(int j=1;j<=k;j++){

if(i<k){

if(j==i+1){

T[i][j] = 1;

}

else{

T[i][j] = 0;

}

continue;

}

//Last Row - store the Coefficients in reverse order

T[i][j] = c[k-j];

}

}

// 3. T^n-1

T = pow(T,n-1);

// 4. multiply by F1

ll res = 0;

for(int i=1;i<=k;i++){

res = (res + (T[1][i]\*F1[i])%MOD)%MOD;

}

return res;

}

int main() {

ll t,n,num;

cin>>t;

while(t--){

cin>>k;

//Init Vector F1

for(int i=0;i<k;i++){

cin>>num;

b.push\_back(num);

}

//Coefficients

for(int i=0;i<k;i++){

cin>>num;

c.push\_back(num);

}

// the value of n

cin>>n;

cout<< compute(n)<<endl;

b.clear();

c.clear();

}

return 0;

}