// NTT 模板题2

#define MAXN 262144

const long long P=50000000001507329LL; // 190734863287 \* 2 ^ 18 + 1

//const int P=1004535809; // 479 \* 2 ^ 21 + 1

//const int P=998244353; // 119 \* 2 ^ 23 + 1

const int G=3;

long long mul(long long x,long long y){

return (x\*y-(long long)(x/(long double)P\*y+1e-3)\*P+P)%P;

}

long long qpow(long long x,long long k,long long p){

long long ret=1;

while(k){

if(k&1) ret=mul(ret,x);

k>>=1;

x=mul(x,x);

}

return ret;

}

long long wn[25];

void getwn(){

for(int i=1; i<=18; ++i){

int t=1<<i;

wn[i]=qpow(G,(P-1)/t,P);

}

}

int len;

void NTT(long long y[],int op){

for(int i=1,j=len>>1,k; i<len-1; ++i){

if(i<j) swap(y[i],y[j]);

k=len>>1;

while(j>=k){

j-=k;

k>>=1;

}

if(j<k) j+=k;

}

int id=0;

for(int h=2; h<=len; h<<=1) {

++id;

for(int i=0; i<len; i+=h){

long long w=1;

for(int j=i; j<i+(h>>1); ++j){

long long u=y[j],t=mul(y[j+h/2],w);

y[j]=u+t;

if(y[j]>=P) y[j]-=P;

y[j+h/2]=u-t+P;

if(y[j+h/2]>=P) y[j+h/2]-=P;

w=mul(w,wn[id]);

}

}

}

if(op==-1){

for(int i=1; i<len/2; ++i) swap(y[i],y[len-i]);

long long inv=qpow(len,P-2,P);

for(int i=0; i<len; ++i) y[i]=mul(y[i],inv);

}

}

void Convolution(long long A[],long long B[],int n){

for(len=1; len<(n<<1); len<<=1);

for(int i=n; i<len; ++i){

A[i]=B[i]=0;

}

NTT(A,1); NTT(B,1);

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

//cout << A[i] << ' ' << B[i] << endl;

A[i]=mul(A[i],B[i]);

}

NTT(A,-1);

}

long long A[MAXN],B[MAXN];

int main(){

getwn();

int t,n;

scanf("%d",&t);

while(t--){

scanf("%d",&n);

long long ans=0;

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

scanf("%lld",&A[i]);

ans+=A[i]\*A[i];

}

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

scanf("%lld",&B[n-i-1]);

ans+=B[n-i-1]\*B[n-i-1];

}

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

A[i+n]=A[i];

B[i+n]=0;

}

/\*

5

1 2 3 4 5

2 3 4 5 1

ans = 0

A[i] 1 2 3 4 5 1 2 3 4 5

B[i] 1 5 4 3 2

**两两相乘**

\sum A\_i^2 + \sum B\_i^2 - 2\sum A\_iB\_{i+k}

把A加长一倍，B反转，构造多项式，这样多项式乘积指数[n,2n)的系数就是各个位置的结果了。

(1 + 2x + 3x^2 + 4x^3 + 5x^4 ) \* (1 + 5x + 4x^2 + 3x^3 + 4x^4)

得到的A[i]: 55 45 40 40 45

\*//\*

相当于

int a[] = {1,2,3,4,5, 1,2,3,4,5};

int b[] = {2,3,4,5,1};

for(int j = 0; j <= 5; j ++)

{

int sum = 0;

for(int i = 0; i <= 4; i++)

sum += a[i+j] \* b[i];

cout << sum << endl;

}

\*/

Convolution(A,B,2\*n);

long long mx=0;

for(int i=n; i<2\*n; ++i){

cout << A[i] << endl;

mx=max(mx,A[i]);

}

printf("%lld\n",ans-2\*mx);