int

```
int add(int x,int y) { return x+y>=moder?x+y-moder:x+y; } int Add(int &x,int y)
{ return x=x+y>=moder?x+y-moder:x+y; }
int sub(int x,int y) { return x<y?x-y+moder:x-y; } int Sub(int &x,int y) {
return x=x<y?x-y+moder:x-y; }
int mul(int x,int y) { return (ll)x*y%moder; } int Mul(int &x,int y) { return x=
(ll)x*y%moder; }
int kuai(int a,int b) { ll rey=1,temp=a; for(;b;b>>=1) { if(b&1)
rey=rey*temp%moder; temp=temp*temp%moder; } return rey; }
void getlim(int n) {
    lim=1,tlim=0; while(lim<=n) lim<<=1,++tlim;
    for(int i=1;i<lim;++i) to[i]=(to[i>>1]>>1)|((i&1)<<(tlim-1));
    return ; }
void ntt(int f[],int mod) {
    int i,j,k,x,y,w1,w;
    for(i=1;i<lim;++i) if(i<to[i]) swap(f[i],f[to[i]]);
    for(i=1;i<lim;i<<=1) {
        w1=kuai(mod==1?3:332748118,(moder-1)/(i<<1));
        for(j=0;j<lim;j+=i<<1)
            for(k=0,w=1;k<i;++k,Mul(w,w1))
                x=f[j|k],y=mul(w,f[i|j|k]),
                f[j|k]=add(x,y),f[i|j|k]=sub(x,y);
    }
    if(mod==-1) { int inv=kuai(lim,moder-2); for(i=0;i<lim;++i) Mul(f[i],inv); }
    return ; }
void Inv(int a[],int b[],int n) {
    if(n==1) return b[0]=kuai(a[0],moder-2),void();
    static int temp[N]={}; Inv(a,b,n+1>>1); int i;
    getlim(n<<1);
    for(i=0;i<n;++i) temp[i]=a[i]; for(i=n;i<lim;++i) temp[i]=0;
    ntt(temp,1),ntt(b,1);
    for(i=0;i<lim;++i) b[i]=mul(b[i],sub(2,mul(b[i],temp[i])));
    ntt(b,-1); for(i=n;i<lim;++i) b[i]=0; return ; }
void Derv(int f[],int ff[],int n) {
    for(int i=1;i<n;++i) ff[i-1]=mul(f[i],i);
    ff[n-1]=0; return ; }
void Inte(int ff[],int f[],int n) {
    for(int i=1;i<n;++i) f[i]=mul(ff[i-1],kuai(i,moder-2));
    f[0]=0; return ;
}
void Ln(int a[],int b[],int n) {
    static int derva[N]={},inva[N]={},dervb[N]={};
    getlim(n<<1); int i;
    for(i=0;i<lim;++i) derva[i]=inva[i]=0;
    Derv(a,derva,n),Inv(a,inva,n);
    ntt(derva,1),ntt(inva,1);
    for(i=0;i<lim;++i) dervb[i]=mul(derva[i],inva[i]);
    ntt(dervb,-1),Inte(dervb,b,n); for(i=n;i<lim;++i) b[i]=0;
    return ; }
void Exp(int a[],int b[],int n) {
    if(n==1) return b[0]=1,void();
    static int lnb[N]={};
    int i; Exp(a,b,n+1>>1); Ln(b,lnb,n);
```

```
    for(i=0;i<n;++i) lnb[i]=sub(a[i],lnb[i]); Add(lnb[0],1);
    getlim(n<<1); ntt(b,1),ntt(lnb,1);
    for(i=0;i<lim;++i) Mul(b[i],lnb[i]);
    ntt(b,-1); for(i=n;i<lim;++i) b[i]=0;
    return ; }
```

SA

```
int oldrk[N]={},id[N]={},cnt[N]={};
struct SA {
    char s[N]={}; int n;
    int rk[N]={},sa[N]={},height[N]={},st[N][LogN]={};
    void setup(int n_,char s_[]) {
        n=n_,memcpy(s,s_,(n+5)*sizeof(char));
        return ;
    }
    void makeSA() {
        int m=128,p,i,j,w,cur;
        for(i=1;i<=n;++i) ++cnt[rk[i]=s[i]];
        for(i=1;i<=m;++i) cnt[i]+=cnt[i-1];
        for(i=n;i;--i) sa[cnt[rk[i]]--]=i;
        for(w=1;;w<<=1,m=p) {
            cur=0;
            for(i=n-w+1;i<=n;++i) id[++cur]=i;
            for(i=1;i<=n;++i) if(sa[i]>w) id[++cur]=sa[i]-w;
            memset(cnt,0,(m+5)<<2);
            for(i=1;i<=n;++i) ++cnt[rk[i]];
            for(i=1;i<=m;++i) cnt[i]+=cnt[i-1];
            for(i=n;i;--i) sa[cnt[rk[id[i]]]--]=id[i];
            p=0;
            memcpy(oldrk,rk,(n+5)<<2);
            for(i=1;i<=n;++i)
                if(oldrk[sa[i]]==oldrk[sa[i-1]]&&oldrk[sa[i]+w]==oldrk[sa[i-
1]+w])
                    rk[sa[i]]=p;
                else rk[sa[i]]=++p;
            if(p==n) break;
        }
        for(i=1,p=0;i<=n;++i) {
            if(!rk[i]) continue;
            if(p) --p;
            while(s[i+p]==s[sa[rk[i]-1]+p]) ++p;
            height[rk[i]]=st[rk[i]][0]=p;
        }
        for(j=1;(1<<j)<=n;++j)
            for(i=1;i+(1<<j)-1<=n;++i)
                st[i][j]=min(st[i][j-1],st[i+(1<<j-1)][j-1]);
        return ;
    }
    int solve(int x,int y) {
        if(x==y) return n-x+1;
        x=rk[x],y=rk[y];
        if(x>y) swap(x,y);
        int k=__lg(y-x);
        return min(st[x+1][k],st[y-(1<<k)+1][k]);
    }
```

```
}
```

poly

```
const int N=500099,P=2000099,moder=998244353; typedef vector<int> poly;
int n,x[N]={},y[N]={},lim,tlim,to[N]={};
int kuai(int a,int b) { ll rey=1,temp=a; for(;b;b>>=1) { if(b&1)
rey=rey*temp%moder; temp=temp*temp%moder; } return rey; }
int add(int x,int y) { return x+y>=moder?x+y-moder:x+y; }
int sub(int x,int y) { return x<y?x-y+moder:x-y; }
int mul(int x,int y) { return (ll)x*y%moder; }
void getlim(int n) {
    lim=1,tlim=0; while(lim<=n) lim<<=1,++tlim;
    for(int i=1;i<lim;++i) to[i]=(to[i>>1]>>1)|((i&1)<<(tlim-1));
    return ; }
void ntt(poly &f,int mod) {
    int i,j,k,w1,w,x,y;
    for(i=1;i<lim;++i) if(i<to[i]) swap(f[i],f[to[i]]);
    for(i=1;i<lim;i<<=1) {
        w1=kuai(mod==1?3:332748118,(moder-1)/(i<<1));
        for(j=0;j<lim;j+=i<<1)
            for(k=0,w=1;k<i;++k,w=mul(w,w1))
                x=f[j|k],y=mul(w,f[i|j|k]),
                f[j|k]=add(x,y),f[i|j|k]=sub(x,y); }
    if(mod==-1) { int inv=kuai(lim,moder-2); for(i=0;i<lim;++i)
f[i]=mul(f[i],inv); }
    return ; }
poly Mul(poly a,poly b) {
    int n=a.size(),m=b.size(),i;
    getlim(n+m-1),a.resize(lim),b.resize(lim),ntt(a,1),ntt(b,1);
    for(i=0;i<lim;++i) a[i]=mul(a[i],b[i]); ntt(a,-1),a.resize(n+m-1);
    return a; }
poly MulT(poly a,poly b) {
    int n=a.size(),m=b.size(),i;
    reverse(b.begin(),b.end()),b=Mul(a,b);
    for(i=0;i<n;++i) a[i]=b[i+m-1];
    return a; }
poly Inv(poly a,int n) {
    if(n==1) return poly(1,kuai(a[0],moder-2));
    static poly temp; int i; poly b=Inv(a,n+1>>1);
    getlim(n<<1),temp.resize(lim),b.resize(lim);
    for(i=0;i<n;++i) temp[i]=a[i];
    for(i=n;i<lim;++i) temp[i]=0;
    ntt(temp,1),ntt(b,1);
    for(i=0;i<lim;++i) b[i]=mul(b[i],sub(2,mul(temp[i],b[i])));
    ntt(b,-1),b.resize(n);
    return b; }
poly Dervt(poly a) {
    int n=a.size(),i; poly b; b.resize(n-1);
    for(i=1;i<n;++i) b[i-1]=mul(a[i],i);
    return b; }
poly G[P]={},G_[P]={};
#define lson (x<<1)
#define rson (x<<1|1)
void evainit(int x,int l,int r,int X[]) {
    if(l==r) { G[x].resize(2),G[x][0]=1,G[x][1]=sub(0,X[l]);
```

```
        G_[x].resize(2),G_[x][0]=sub(0,X[l]),G_[x][1]=1; return ; }
    int mid=l+r>>1; evainit(lson,l,mid,X),evainit(rson,mid+1,r,X);
    G[x]=Mul(G[lson],G[rson]),G_[x]=Mul(G_[lson],G_[rson]); return ; }
void evadg(int x,int l,int r,poly f,int v[]) {
    f.resize(r-l+1);
    if(l==r) { v[l]=f[0]; return ; }
    int mid=l+r>>1;
    evadg(lson,l,mid,MulT(f,G[rson]),v);
    evadg(rson,mid+1,r,MulT(f,G[lson]),v);
    return ; }
void eva(poly f,int X[],int v[],int n) {
    f.resize(n+1),evainit(1,1,n,X);
    evadg(1,1,n,MulT(f,Inv(G[1],n+1)),v);
    return ; }
int _delta[N]={};
poly interdg(int x,int l,int r,int Y[]) {
    if(l==r) return poly(1,mul(Y[l],kuai(_delta[l],moder-2)));
    int mid=l+r>>1,i;
    poly L=interdg(lson,l,mid,Y),R=interdg(rson,mid+1,r,Y);
    L=Mul(L,G_[rson]),R=Mul(R,G_[lson]);
    for(i=0;i<R.size();++i) L[i]=add(L[i],R[i]);
    return L; }
poly inter(int X[],int Y[],int n) {
    evainit(1,1,n,X);
    poly delta=G_[1]; delta=Dervt(delta);
    evadg(1,1,n,MulT(delta,Inv(G[1],n+1)),_delta);
    return interdg(1,1,n,Y); }
poly ans;
int main()
{
//  usefile("inter");
    int i;
    read(n);
    for(i=1;i<=n;++i) read(x[i],y[i]);
    ans=inter(x,y,n); for(i=0;i<n;++i) printf("%d ",ans[i]);
    printf("\n"); return 0;
}
```