FFT迭代

```
#include <bits/stdc++.h>
using namespace std;
const int maxn=1e7+10;
const double Pi=acos(-1.0);
inline int read(){
    char c=getchar();int x=0,f=1;
    while (c<'0'||c>'9'){if(c=='-') f=-1;c=getchar();}
    while (c>='0'&&c<='9'){x=x*10+c-'0';c=getchar();}
    return x*f;
}
struct CP{
    double x,y;
    CP (double xx=0,double yy=0){x=xx,y=yy;}
}a[maxn],b[maxn];
CP operator + (CP a,CP b){return CP(a.x+b.x,a.y+b.y);}
CP operator - (CP a,CP b){return CP(a.x-b.x,a.y-b.y);}
CP operator * (CP a,CP b){return CP(a.x*b.x-a.y*b.y,a.x*b.y+a.y*b.x);}
int n,m,l,r[maxn],limit=1;
void FFT(CP *A,int type){
    for(int i=0;i<limit;i++) if(i<r[i]) swap(A[i],A[r[i]]);
    for(int mid=1;mid<limit;mid<<=1){
        CP Wn(cos(Pi/mid),type*sin(Pi/mid));
        for(int R=mid<<1,j=0;j<limit;j+=R){
            CP w(1,0);
            for(int k=0;k<mid;k++,w=w*Wn){
                CP x=A[j+k],y=w*A[j+mid+k];
                A[j+k]=x+y;
                A[j+mid+k]=x-y;
            }
        }
    }
}
int main(){
    int n=read(),m=read();
    for(int i=0;i<=n;i++) a[i].x=read();
    for(int i=0;i<=m;i++) b[i].x=read();
    while (limit<=n+m) limit<<=1,l++;
    for(int i=0;i<limit;i++) r[i]=(r[i>>1]>>1)|((i&1)<<(l-1));
    FFT(a,1);
    FFT(b,1);
    for(int i=0;i<=limit;i++) a[i]=a[i]*b[i];
    FFT(a,-1);
    for(int i=0;i<=n+m;i++) printf("%d ",(int)(a[i].x/limit+0.5));
    return 0;
}
```