Pastebin Link: <https://paste.ubuntu.com/p/nZ7zYBGyk3/>

*/\**  
 *Call Multiply to multiply two polynomials.*  
 *Vectors a and b hold the co-efficients of the polynomials.*  
 *Vector res will contain the co-efficients of a(x)\*b(x)*  
  
 *Call multiply\_two\_numbers, to multiply two strings*  
*\*/*  
  
#define PI acos(-1.0)  
**using** **namespace** std;  
**typedef** complex <**double**> base;  
**void** fft(vector <base> &a, **bool** invert){  
 **int** n=(**int**) a.size();  
 **if**(n==1) **return**;  
  
 vector <base> a0(n>>1), a1(n>>1);  
 **for**(**int** i=0, j=0; i<n; i+=2, j++){  
 a0[j]=a[i];  
 a1[j]=a[i+1];  
 }  
 fft(a0, invert);  
 fft(a1, invert);  
  
 **double** ang=2\*PI/n \* (invert? -1: 1);  
 base w(1), wn(cos(ang), sin(ang));  
 **for**(**int** i=0; i<(n>>1); i++){  
 a[i]=a0[i]+w\*a1[i];  
 a[i+(n>>1)]=a0[i]-w\*a1[i];  
 **if**(invert) a[i]/=2, a[i+(n>>1)]/=2;  
 w\*=wn;  
 }  
}  
  
**void** multiply(**const** vector <**int**> &a, **const** vector <**int**> &b, vector <**int**> &res){  
 vector <base> fa(a.begin(), a.end()), fb(b.begin(), b.end());  
 **int** n=1;  
 **while**(n<max(a.size(), b.size())) n<<=1;  
 n<<=1;  
 fa.resize(n);  
 fb.resize(n);  
 fft(fa, **false**);  
 fft(fb, **false**);  
 **for**(**int** i=0; i<n; i++) fa[i]\*=fb[i];  
 fft(fa, **true**);  
 res.resize(n);  
 **for**(**int** i=0; i<n; i++) res[i]=**int**(fa[i].real()+0.5);  
}  
  
**void** multiply\_two\_numbers(**char**\* an, **char**\* bn, **int** al, **int** bl, vector <**int**> &res){  
 vector <**int**> a, b;  
 **for**(**int** i=0; i<al; i++) a.push\_back(**int**(an[i]-'0'));  
 **for**(**int** i=0; i<bl; i++) b.push\_back(**int**(bn[i]-'0'));  
 reverse(a.begin(), a.end());  
 reverse(b.begin(), b.end());  
 multiply(a, b, res);  
 **int** carry=0;  
 **for**(**int** i=0; i<(**int**)res.size(); i++){  
 res[i]+=carry;  
 carry=res[i]/10;  
 res[i]%=10;  
 }  
 **if**(carry) res.push\_back(carry);  
}