# FFT

5730

#include<cstdio>

#include<cstring>

#include<algorithm>

#include<cmath>

#include<set>

#include<iostream>

using namespace std;

typedef unsigned long long ULL;

const int M = 4e+5+123;

const int mod=313;

//fft模板 ，必要时需要讲double全部替换为long double

double PI=acos(-1.0);

struct Complex{

double real, image;

Complex(double \_real, double \_image){

real = \_real,image = \_image;

}

Complex(){}

};

Complex operator + (const Complex &c1, const Complex &c2){

return Complex(c1.real + c2.real, c1.image + c2.image);

}

Complex operator - (const Complex &c1, const Complex &c2){

return Complex(c1.real - c2.real, c1.image - c2.image);

}

Complex operator \* (const Complex &c1, const Complex &c2){

return Complex(c1.real\*c2.real - c1.image\*c2.image, c1.real\*c2.image + c1.image\*c2.real);

}

int rev(int id, int len){

int ret = 0;

for(int i = 0; (1 << i) < len; i++){

ret <<= 1;

if(id & (1 << i)) ret |= 1;

}

return ret;

}

Complex U[M\*2];

void FFT(Complex\* a, int len, int DFT){

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

U[rev(i, len)] = a[i];

for(int s = 1; (1 << s) <= len; s++){

int m = (1 << s);

Complex wm = Complex(cos(DFT\*2\*PI/m), sin(DFT\*2\*PI/m));

for(int k = 0; k < len; k += m){

Complex w = Complex(1, 0);

for(int j = 0; j < (m >> 1); j++){

Complex t = w\*U[k + j + (m >> 1)];

Complex u = U[k + j];

U[k + j] = u + t;

U[k + j + (m >> 1)] = u - t;

w = w\*wm;

}

}

}

if(DFT == -1) for(int i = 0; i < len; i++) U[i].real /= len, U[i].image /= len;

for(int i = 0; i < len; i++) a[i] = U[i];

return;

}

long long a[M];

long long f[M];

Complex A[M],B[M],X[M];

int cdq(int l,int r){

if(l==r){

f[l]+=a[l];

f[l]%=mod;

return 0;

}

int mid=(l+r)>>1;

cdq(l,mid);

int Alen=mid-l+1;

int Blen=r-l+1;

for(int i=l;i<=mid;i++){

A[i-l]=Complex(f[i],0);

}

for(int i=l;i<=r;i++){

B[i-l]=Complex(a[i-l],0);

}

int len=1;

while(len<=Alen+Blen) len=len\*2;

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

A[i]=Complex(0,0);

}

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

B[i]=Complex(0,0);

}

FFT(A,len,1);

FFT(B,len,1);

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

X[i]=A[i]\*B[i];

}

FFT(X,len,-1);

for(int i=mid+1;i<=r;i++){

long long tmp=((int)(X[i-l].real+0.5))%mod;

f[i]=(f[i]+tmp)%mod;

}

cdq(mid+1,r);

return 0;

}

int main(){

int n;

while(cin>>n&&n){

a[0]=1;

for(int i=1;i<=n;i++)

cin>>a[i],f[i]=0,a[i]=a[i]%mod;

cdq(1,n);

printf("%lld\n",f[n]%mod);

}

return 0;

}