

polyn(FFT,NTT)

```
1  #include <bits/stdc++.h>
2  #define int long long
3  #define endl '\n'
4  #define LL __int128
5  using namespace std;
6  int qpow(int a, int b, int p) {int ret = 1; for(a %= p; b; b >>= 1, a = a *
a % p) if(b & 1) ret = ret * a % p; return ret; }
7  int qpow(int a,int b) {int ret = 1; for(; b; b >>= 1, a *= a) if(b & 1) ret
*= a; return ret; }
8  int gcd(int x,int y) {return y ? gcd(y, x % y) : x; }
9  pair<int,int> exgcd(int a,int b) { if(!b) return {1, 0}; pair<int,int> ret
= exgcd(b, a % b); return {ret.second, ret.first - a / b * ret.second }; }
10 int lcm(int x,int y){ return x / gcd(x, y) * y; }
11 namespace polyn {
12     const int N = 1 << 21;
13     const int P = 998244353;
14     const int G = 3;
15     const int GI = 332748118;
16     int inv;
17     int rev[N];
18     struct Z {
19         double a, b;
20         Z(const double &a = 0, const double &b = 0) {
21             a = _a, b = _b;
22         }
23         Z operator + (const Z &x) {
24             return Z(a + x.a, b + x.b);
25         }
26         Z operator - (const Z &x) {
27             return Z(a - x.a, b - x.b);
28         }
29         Z operator * (const Z &x) {
30             return Z(a * x.a - b * x.b, a * x.b + b * x.a);
31         }
32     };
33     const double PI = acos(-1);
34     void init(const int &n, const int &pw) {
35         for (int i = 0; i < n; i++) {
36             rev[i] = (rev[i >> 1] >> 1) | ((i & 1) << (pw - 1));
37         }
38         inv = qpow(n, P - 2, P);
39     }
40     void FFT(Z *z, int n, bool T) {
41         for (int i = 0; i < n; i++) {
42             if (rev[i] > i) {
43                 swap(z[rev[i]], z[i]);
44             }
45         }
46         for (int len = 2; len <= n; len <<= 1) {
47             Z wn = Z(cos(2.0 * PI / n), (T ? 1 : -1) * sin(2.0 * PI / n));
48             for (int l = 0, r = len - 1; r < n; l += len, r += len) {
49                 Z w = Z(1, 0);
```

```

50         for (int k = 1, mid = 1 + (len >> 1); k < mid; k++, w = w *
wn) {
51             z[x] = z[k], y = w * z[k + (len >> 1)];
52             z[k] = x + y, z[k + (len >> 1)] = x - y;
53         }
54     }
55 }
56 if (T) return;
57 for (int i = 0; i < n; i++) {
58     z[i].a /= n, z[i].b /= n;
59 }
60 }
61 void NTT(int *z, int n, bool T) {
62     for (int i = 0; i < n; i++) {
63         if (rev[i] > i) {
64             swap(z[rev[i]], z[i]);
65         }
66     }
67     for (int len = 2; len <= n; len <= 1) {
68         int wn = qpow(T ? G : GI, (P - 1) / len, P);
69         for (int l = 0, r = len - 1; r < n; l += len, r += len) {
70             int w = 1;
71             for (int k = 1, mid = 1 + (len >> 1); k < mid; k++, w = w *
wn % P) {
72                 int x = z[k], y = w * z[k + (len >> 1)] % P;
73                 z[k] = (x + y) % P, z[k + (len >> 1)] = (x - y + P) %
P;
74             }
75         }
76     }
77     if (T) return;
78     for (int i = 0; i < n; i++) {
79         z[i] = z[i] * inv % P;
80     }
81 }
82 }
83 const int mod = 998244353;
84 const int _N = 1 << 21;
85 int _a[_N], _b[_N];
86 void cdq(int *f, int *g, int l, int r) {
87     if (l == r) {
88         return;
89     }
90     int mid = l + r >> 1;
91     cdq(f, g, l, mid);
92     for (int i = l; i <= mid; i++) {
93         _a[i - l] = f[i];
94     }
95     for (int i = mid + 1; i <= r; i++) {
96         _a[i - l] = 0;
97     }
98     for (int i = l; i <= r; i++) {
99         _b[i - l] = g[i - l];
100     }
101     int n = 1, pw = 0;
102     while (n < r - l + 1) {
103         n <= 1, pw++;
104     }

```

```

105     polyn :: init(n, pw);
106     polyn :: NTT(_a, n, 1);
107     polyn :: NTT(_b, n, 1);
108     for (int i = 0; i < n; i++) {
109         _a[i] = _a[i] * _b[i] % polyn :: P;
110     }
111     polyn :: NTT(_a, n, 0);
112     for (int i = mid + 1; i <= r; i++) {
113         f[i] += _a[i - 1];
114         f[i] %= mod;
115     }
116     cdq(f, g, mid + 1, r);
117 }
118 const int N = 5 + 1e6;
119 int g[N], f[N];
120 signed main() {
121     ios :: sync_with_stdio(false), cin.tie(0), cout.tie(0);
122     int n; cin >> n;
123     for (int i = 1; i < n; i++) {
124         cin >> g[i];
125     }
126     f[0] = 1;
127     int _n = 1;
128     while (_n < n) {
129         _n <<= 1;
130     }
131     cdq(f, g, 0, _n);
132     for (int i = 0; i < n; i++) {
133         cout << f[i] << " ";
134     }
135     cout << endl;
136 }

```