```cpp
#include <bits/stdc++.h>

using namespace std;
using ll = long long;

long long gcd(long long a, long long b) {
  if (a % b == 0)
    return b;
  return gcd(b, a % b);
}

// return (x, y) (a * x + b * y = gcd(a, b))
pair<long long, long long> exeuclid(long long a, long long b) {
  if (b == 0)
    return make_pair(1, 0);
  pair<long long, long long> ret = exeuclid(b, a % b);
  ret.first -= a / b * ret.second;
  return make_pair(ret.second, ret.first);
}

// v := [(modulo, reminder)]
// moduloes should be coprime
long long CRT(vector<pair<ll, ll>> v) {
  long long mod = 1;
  long long ret = 0;
  for (int i = 0; i < v.size(); i++) {
    pair<ll, ll> xy = exeuclid(mod, v[i].first);
    long long M = mod * v[i].first;
    long long l = ((v[i].second * mod) % M) * xy.first;
    l %= M;
    long long r = ((ret * v[i].first) % M) * xy.second;
    r %= M;
    ret = (l + r) % M;
    ret += M;
    ret %= M;
    mod = M;
  }
  return ret;
}

ll mod_pow(ll e, ll x, ll p) {
  ll r = 1;
  e %= p;
  while (x) {
    if (x & 1) {
      r *= e;
      r %= p;
    }
    e *= e;
    e %= p;
```

```
  51     x >>= 1;
  52   }
  53   return r;
  54 }
  55
  56 template<class T>
  57 T extgcd(T a, T b, T &x, T &y) {
  58   for (T u = y = 1, v = x = 0; a;) {
  59     T q = b / a;
  60     swap(x -= q * u, u);
  61     swap(y -= q * v, v);
  62     swap(b -= q * a, a);
  63   }
  64   return b;
  65 }
  66
  67 ll mod_inv(ll a, ll b) {
  68   return (exeuclid(a, b).first % b + b) % b;
  69 }
  70
  71 template<int mod, int primitive_root>
  72 class NTT {
  73 public:
  74   int get_mod() const { return mod; }
  75
  76   void _ntt(vector<ll> &a, int sign) {
  77     const int n = a.size();
  78     assert((n ^ (n & -n)) == 0); //n = 2^k
  79
  80     const int g = 3; //g is primitive root of mod
  81     int h = (int) mod_pow(g, (mod - 1) / n, mod); // h^n = 1
  82     if (sign == -1) h = (int) mod_inv(h, mod); //h = h^-1 % mod
  83
  84     //bit reverse
  85     int i = 0;
  86     for (int j = 1; j < n - 1; ++j) {
  87       for (int k = n >> 1; k > (i ^= k); k >>= 1);
  88       if (j < i) swap(a[i], a[j]);
  89     }
  90
  91     for (int m = 1; m < n; m *= 2) {
  92       const int m2 = 2 * m;
  93       const ll base = mod_pow(h, n / m2, mod);
  94       ll w = 1;
  95       for (int x = 0; x < m; x++) {
  96         for (int s = x; s < n; s += m2) {
  97           ll u = a[s];
  98           ll d = a[s + m] * w % mod;
  99           a[s] = u + d;
 100           if (a[s] >= mod) a[s] -= mod;
```

```
101           a[s + m] = u - d;
102           if (a[s + m] < 0) a[s + m] += mod;
103         }
104         w = w * base % mod;
105       }
106     }
107
108     for (auto &x : a) if (x < 0) x += mod;
109   }
110
111   void ntt(vector<ll> &input) {
112     _ntt(input, 1);
113   }
114
115   void intt(vector<ll> &input) {
116     _ntt(input, -1);
117     const int n_inv = mod_inv(input.size(), mod);
118     for (auto &x : input) x = x * n_inv % mod;
119   }
120
121   // 畳み込み演算を行う
122   vector<ll> convolution(const vector<ll> &a, const vector<ll> &b) {
123     int ntt_size = 1;
124     while (ntt_size < a.size() + b.size()) ntt_size *= 2;
125
126     vector<ll> _a = a, _b = b;
127     _a.resize(ntt_size);
128     _b.resize(ntt_size);
129
130     ntt(_a);
131     ntt(_b);
132
133     for (int i = 0; i < ntt_size; i++) {
134       (_a[i] *= _b[i]) %= mod;
135     }
136
137     intt(_a);
138     return _a;
139   }
140 };
141
142 ll garner(vector<pair<ll, ll>> mr, int mod) {
143   mr.emplace_back(mod, 0);
144
145   vector<ll> coffs(mr.size(), 1);
146   vector<ll> constants(mr.size(), 0);
147   for (int i = 0; i < mr.size() - 1; i++) {
148     // coffs[i] * v + constants[i] == mr[i].second (mod mr[i].first) を解く
149     ll v = (mr[i].second - constants[i]) * mod_inv(coffs[i], mr[i].first) % mr[i].first;
150     if (v < 0) v += mr[i].first;
```

```
151
152     for (int j = i + 1; j < mr.size(); j++) {
153       (constants[j] += coffs[j] * v) %= mr[j].first;
154       (coffs[j] *= mr[i].first) %= mr[j].first;
155     }
156   }
157
158   return constants[mr.size() - 1];
159 }
160
161 typedef NTT<167772161, 3> NTT_1;
162 typedef NTT<469762049, 3> NTT_2;
163 typedef NTT<1224736769, 3> NTT_3;
164 NTT_1 ntt1;
165 NTT_2 ntt2;
166 NTT_2 ntt3;
167 // ref: https://math314.hateblo.jp/entry/2015/05/07/014908
168
169 // NTT
170 // c[i] = for j,k a[j] + b[k] if j + k == i mod P
171 vector<ll> mod_conv(vector<ll> a, vector<ll> b, ll P) {
172   for (auto &i : a) i %= P;
173   for (auto &i : b) i %= P;
174   auto v1 = ntt1.convolution(a, b);
175   auto v2 = ntt2.convolution(a, b);
176   auto v3 = ntt3.convolution(a, b);
177   int n = v2.size();
178   vector<ll> ret(n);
179   for (int i = 0; i < n; i++) {
180     if (v1[i] != v2[i]) cerr << v1[i] << " " << v2[i] << endl;
181     ret[i] = garner(vector<pair<ll, ll>>(
182         {
183           make_pair(ntt1.get_mod(), v1[i]),
184           make_pair(ntt2.get_mod(), v2[i]),
185           make_pair(ntt3.get_mod(), v3[i])
186         }), P);
187   }
188   return ret;
189 }
190
191 int main() {
192   int n;
193   cin >> n;
194   vector<ll> a(n + 1), b(n + 1);
195   for (int i = 1; i <= n; i++) {
196     cin >> a[i] >> b[i];
197   }
198   auto c = mod_conv(a, b, 1000000007);
199   for (int i = 1; i <= 2 * n; i++) {
200     cout << c[i] << endl;
```

```
201   }
202   return 0;
203 }
```