Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0359 – Перепечаев Виктор Александрович

A	B	C	D	E	F	sum
100	100	100	35	20	90	445

Task A

```
//                  (     ):  24fin0359

//       :  offecreen // #pragma GCC optimize("O3")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=
    native")

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
//using ull = unsigned ll;
using ld = long double;

// DEFINES
#define EACH(el, v) for (auto &el : v)
#define SZ(x) (int)x.size()
#define ALL(x) x.begin(), x.end()

// DEBUG
```

```
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
#endif

const int N = 100;

int DP[N][N];

void solve() {
  int a, b, n;
  cin >> a >> b >> n;
  if (a > b) swap(a, b);
  for (int i  = 0; i < n; ++i) {
    int r = i + a - 1;
    if (r >= n) continue;
    DP[i][r] = 1;
  }
  for (int i = 0; i < n; ++i) {
    int r = i + b - 1;
    if (r >= n) continue;
    DP[i][r] += 1;
  }
  for (int l = 0; l < n; ++l) {
    DP[l][l + a - 1] = 1;
    DP[l][l + b - 1] + 1;
    for (int r = a; r < n; ++r) {
      DP[l][r];
      if (r - a >= l)
        DP[l][r] += DP[l][r - a];
      if (r - b >= l)
        DP[l][r] += DP[l][r - b];
    }
  }
  // for (int l = 0; l < n; ++l) {
  //  for (int r = l; r < n; ++r) {
  //    cout << "DP[" << l << "][" << r << "] = " << DP[l][r] << '\n';
  //  }
  // }
  int ANS = 0;
  for (int i = n - a; i < n; ++i) {
    if (a > n) continue;
    int r = i - 1;
    int l = (i + a) % n;
    // debug(l);
    // debug(r);
    if (i == n - 1 && i - a + 1 == 0) {
      ANS++;
      continue;
```

```
    }
    if (l == r + 1) {
      ANS++;
      continue;
    }
    if (!(0 <= l && l < n && 0 <= r && r < n)) continue;
    if (l > r) continue;
    ANS += DP[l][r];
  }
  for (int i = n - b; i < n; ++i) {
    if (b > n) continue;
    int r = i - 1;
    int l = (i + b) % n;
    // debug(l);
    // debug(r);
    if (i == n - 1 && i - b + 1 == 0) {
      ANS++;
      continue;
    }
    if (l == r + 1) {
      ANS++;
      continue;
    }
    if (!(0 <= l && l < n && 0 <= r && r < n)) continue;
    if (l > r) continue;
    ANS += DP[l][r];
  }
  cout << ANS;
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int tests = 1;
    // cin >> tests;
    while (tests--) {
        solve();
    }
    return 0;
}
```

Task B

```
//                    (      ):  24fin0359

//        :  offecreen // #pragma GCC optimize("O3")
```

```
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=
    native")

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
//using ull = unsigned ll;
using ld = long double;

// DEFINES
#define EACH(el, v) for (auto &el : v)
#define SZ(x) (int)x.size()
#define ALL(x) x.begin(), x.end()

// DEBUG
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
#endif

struct P {
  int x, y, z;
  P() {
    x = 0;
    y = 0;
    z = 0;
  }
  P(int a, int b, int c) {
    x = a;
    y = b;
    z = c;
  }
  bool operator < (const P &other) const {
    if (x != other.x) return x < other.x;
    if (y != other.y) return y < other.y;
    return z < other.z;
  }
  bool operator == (const P &other) const {
```

```
    return x == other.x && y == other.y && z == other.z;
  }
};

map<P, int> D;

vector<P> SDV {
  P(1, 0, -1),
  P(1, -1, 0),
  P(0, -1, 1),
  P(-1, 0, 1),
  P(-1, 1, 0),
  P(0, 1, -1)
};

int n;

// P inv(P p) {
//   if (p.x == n || p.x == -n) {
//     return P(-p.x, -p.z, -p.y);
//   }
//   if (p.y == n || p.y == -n) {
//     return P(-p.z, -p.y, -p.x);
//   }
//   return P(-p.y, -p.x, -p.z);
// }

P inv_x(P p) {
  return P(-p.x, -p.z, -p.y);
}

P inv_y(P p) {
  return P(-p.z, -p.y, -p.x);
}

P inv_z(P p) {
  return P(-p.y, -p.x, -p.z);
}

int dist(P S, P F) {
  return max({abs(S.x - F.x), abs(S.y - F.y), abs(S.z - F.z)});
}

void solve() {
  cin >> n;
  P S, F;
  cin >> S.x >> S.y >> S.z;
  cin >> F.x >> F.y >> F.z;
  int s1 = max({abs(S.x - F.x), abs(S.y - F.y), abs(S.z - F.z)});
  P at(n, 0, -n);
  s1 = min(s1, min(dist(S, at) + dist(inv_z(at), F), dist(S, inv_z(at)) +
     dist(at, F)));
  for (int i = 0; i <= n; ++i) {
```

```
  s1 = min(s1, min(dist(S, at) + dist(inv_x(at), F), dist(S, inv_x(at))
   + dist(at, F)));
  // if (s1 == 3) {
  //  debug(at.x);
  //  debug(at.y);
  //  debug(at.z);
  // }
  at.y--;
  at.z++;
}
//////////////
at = P(n, -n, 0);
s1 = min(s1, min(dist(S, at) + dist(inv_x(at), F), dist(S, inv_x(at)) +
   dist(at, F)));
for (int i = 0; i <= n; ++i) {
  s1 = min(s1, min(dist(S, at) + dist(inv_y(at), F), dist(S, inv_y(at))
   + dist(at, F)));
  at.x--;
  at.z++;
}
///////////////
at = P(0, -n, n);
s1 = min(s1, min(dist(S, at) + dist(inv_y(at), F), dist(S, inv_y(at)) +
   dist(at, F)));
for (int i = 0; i <= n; ++i) {
  s1 = min(s1, min(dist(S, at) + dist(inv_z(at), F), dist(S, inv_z(at))
   + dist(at, F)));
  at.x--;
  at.y++;
}
////////////
// at = P(-n, 0, n);
// s1 = min(s1, min(dist(S, at) + dist(inv_x(at), F), dist(S, inv_x(at)
  ) + dist(at, F)));
// for (int i = 0; i <= n; ++i) {
//  s1 = min(s1, min(dist(S, at) + dist(inv(at), F), dist(S, inv(at)) +
   dist(at, F)));
//  at.y++;
//  at.z--;
// }
// //////////
// at = P(-n, n, 0);
// for (int i = 0; i <= n; ++i) {
//  s1 = min(s1, min(dist(S, at) + dist(inv(at), F), dist(S, inv(at)) +
   dist(at, F)));
//  at.x++;
//  at.z--;
// }
// //////////
// at = P(0, n, -n);
// for (int i = 0; i <= n; ++i) {
//  s1 = min(s1, min(dist(S, at) + dist(inv(at), F), dist(S, inv(at)) +
   dist(at, F)));
```

```
//  at.x++;
//  at.y--;
// }
cout << s1;
// if (n > 10) {
//  cout << s1;
//  return;
// }
// deque<P> q;
// D[S] = 1;
// q.push_front(S);
// while (!q.empty()) {
//  P at = q.front();
//  debug(at.x);
//  debug(at.y);
//  debug(at.z);
//  debug(D[at]);
//  cout << "------" << endl;
//  if (at == F) break;
//  q.pop_front();
//  if (at.x == n || at.x == -n) {
//    int to_x, to_y, to_z;
//    to_x = -at.x;
//    to_y = -at.y;
//    to_z = -at.z;
//    swap(to_y, to_z);
//    if (D[P(to_x, to_y, to_z)] != 0) continue;
//    D[P(to_x, to_y, to_z)] = D[at];
//    q.push_front(P(to_x, to_y, to_z));
//  } else if (at.y == n || at.y == -n) {
//    int to_x, to_y, to_z;
//    to_y = -at.y;
//    to_x = -at.x;
//    to_z = -at.z;
//    swap(to_x, to_z);
//    if (D[P(to_x, to_y, to_z)] != 0) continue;
//    D[P(to_x, to_y, to_z)] = D[at];
//    q.push_front(P(to_x, to_y, to_z));
//  } else if (at.z == n || at.z == -n) {
//    int to_x, to_y, to_z;
//    to_y = -at.y;
//    to_x = -at.x;
//    to_z = -at.z;
//    swap(to_x, to_y);
//    if (D[P(to_x, to_y, to_z)] != 0) continue;
//    D[P(to_x, to_y, to_z)] = D[at];
//    q.push_front(P(to_x, to_y, to_z));
//  }
//  for (auto [x, y, z] : SDV) {
//    int to_x, to_y, to_z;
//    bool did = false;
//    to_x = at.x + x;
//    to_y = at.y + y;
```

```
  //    to_z = at.z + z;
  //    if (abs(to_x) > n || abs(to_y) > n || abs(to_z) > n) continue;
  //    if (D[P(to_x, to_y, to_z)] != 0) continue;
  //    D[P(to_x, to_y, to_z)] = D[at] + 1;
  //    if (P(to_x, to_y, to_z) == F) break;
  //    q.push_back(P(to_x, to_y, to_z));
  //  }
  // }
  // cout << D[F] - 1;
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int tests = 1;
    // cin >> tests;
    while (tests--) {
        solve();
    }
    return 0;
}
```

Task C

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace __gnu_pbds;

using namespace std;

// #pragma GCC optimize("Ofast,unroll-loops")
// #pragma GCC target("avx2,tune=native")

// #define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
using ull = unsigned ll;
using ld = long double;

// DEFINES
#define all(x) x.begin(), x.end()
```

```
// DEBUG
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
#endif

const int N = 1e5 + 228;

const int BASE = 179, MOD1 = 1e9 + 1329, MOD2 = 1e9 + 7;

const int symbol = '0';

struct Mint {
    int h1, h2;
    Mint() {
        h1 = h2 = 0;
    }
    Mint(char c) {
        h1 = c - symbol + 1;
        h2 = c - symbol + 1;
    }
    Mint(int x) {
        h1 = h2 = x;
    }
    Mint(int x1, int x2) {
        h1 = x1;
        h2 = x2;
    }
    Mint operator * (Mint other) {
        return Mint(1ll * h1 * other.h1 % MOD1, 1ll * h2 * other.h2 %
    MOD2);
    }
    Mint operator * (int x) {
        return Mint(1ll * h1 * x % MOD1, 1ll * h2 * x % MOD2);
    }
    Mint operator - (Mint other) {
        int r1, r2;
        if (h1 - other.h1 < 0) r1 = h1 - other.h1 + MOD1;
        else r1 = h1 - other.h1;
        if (h2 - other.h2 < 0) r2 = h2 - other.h2 + MOD2;
        else r2 = h2 - other.h2;
        return Mint(r1, r2);
    }
    Mint operator + (Mint other) {
        int r1, r2;
        if (h1 + other.h1 >= MOD1) r1 = h1 + other.h1 - MOD1;
        else r1 = h1 + other.h1;
        if (h2 + other.h2 >= MOD2) r2 = h2 + other.h2 - MOD2;
```

```
        else r2 = h2 + other.h2;
        return Mint(r1, r2);
    }
    bool operator == (Mint other) {
        return h1 == other.h1 && h2 == other.h2;
    }
};

Mint P[N];

void precalc() {
    P[0] = Mint(1);
    for (int i = 1; i < N; ++i) {
        P[i] = P[i - 1] * BASE;
    }
}

vector<Mint> V_Hash(string &s) {
    int n = s.size();
    vector<Mint> res(n);
    res[0] = Mint(s[0]);
    for (int i = 1; i < n; ++i) {
        res[i] = res[i - 1] * BASE + Mint(s[i]);
    }
    return res;
}

Mint get_hash(int l, int r, vector<Mint> &hhhash) {
    if (l == 0) return hhhash[r];
    return Mint(hhhash[r] - hhhash[l - 1] * P[r - l + 1]);
}

void solve() {
    precalc();
    string t;
  cin >> t;
  if (t[0] == 'f') {
    int n;
    string s;
    cin >> n >> s;
    Mint hash = V_Hash(s).back();
    cout << hash.h1 << '#' << hash.h2;
    cout.flush();
  } else {
    char ccc;
    int n;
    Mint orig;
    string new_s;
    cin >> n >> new_s >> orig.h1 >> ccc >> orig.h2;
    new_s = new_s + new_s;
    vector<Mint> new_s_hash = V_Hash(new_s);
    int new_sn = 2 * n;
    for (int i = 0; i < 2 * n; ++i) {
```

```
      int r = i + n - 1;
      if (r >= new_sn) continue;
      if (orig == get_hash(i, r, new_s_hash)) {
        cout << "YES";
        cout.flush();
        return;
      }
    }
    cout << "NO";
    cout.flush();
  }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int times = 1;
    // cin >> times;
    while (times--) {
        solve();
    }
    return 0;
}
```

Task D

```
//                  (     ):  24fin0359

//       :  offecreen // #pragma GCC optimize("O3")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=
    native")

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
//using ull = unsigned ll;
```

```
using ld = long double;

// DEFINES
#define EACH(el, v) for (auto &el : v)
#define SZ(x) (int)x.size()
#define ALL(x) x.begin(), x.end()

// DEBUG
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
#endif

int get_rand(int l, int r) {
  return rnd() % (r - l + 1) + l;
}

void solve() {
  int n;
  cin >> n;
  int t;
  if (n == 3) t = 3 * 2 * 1;
  if (n == 4) t = 4 * 3 * 2 * 1;
  if (n == 5) t = 5 * 4 * 3 * 2 * 1;
  while (true) { //
    for (int i = 0; i < n; ++i) {
      cout << 1 << ' ' << 1 << '\n';
      cout.flush();
      char res;
      cin >> res;
      if (res == '!') {
        return;
      }
    }
    int i = get_rand(1, n);
    int j = get_rand(1, n);
    cout << i << ' ' << j << '\n';
    cout.flush();
    char res;
    cin >> res;
    if (res == '!') {
      return;
    }
  }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
```

```
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int tests = 1;
    // cin >> tests;
    while (tests--) {
        solve();
    }
    return 0;
}
```

Task E

```
//                  (     ):  24fin0359

//       :  offecreen // #pragma GCC optimize("O3")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=
    native")

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
//using ull = unsigned ll;
using ld = long double;

// DEFINES
#define EACH(el, v) for (auto &el : v)
#define SZ(x) (int)x.size()
#define ALL(x) x.begin(), x.end()

// DEBUG
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
```

```
#endif

const int N = 1e5 + 228;

vector<int> G[N];

int T[N], CNT[N];

void solve() {
  int n, q;
  cin >> n >> q;
  for (int i = 0; i < n; ++i) {
    int d;
    cin >> d;
    G[i].resize(d);
    for (int j = 0; j < d; ++j) {
      cin >> G[i][j];
      G[i][j]--;
    }
  }
  int at = 0;
  for (int t = 0; t <= 1e5; ++t) {
    T[t] = at;
    int idx = CNT[at] % G[at].size();
    CNT[at]++;
    at = G[at][idx];
  }
  vector<int> qs(q);
  bool ok = true;
  for (int i = 0; i < q; ++i) {
    cin >> qs[i];
    if (qs[i] > 1e5) ok = false;
  }
  if (ok) {
    for (auto t : qs) {
      cout << T[t] + 1 << ' ';
    }
    return;
  } else {
    debug(n);
    int CNT_OK = 0;
    fill(CNT, CNT + N, 0);
    int cnt = 0;
    int at = 0;
    int start = G[at][0];
    vector<int> TTT(1e7);
    do {
      TTT[cnt] = at;
      int idx = CNT[at] % G[at].size();
      CNT[at]++;
      at = G[at][idx];
      cnt++;
      idx = CNT[at] % G[at].size();
```

```
      if (idx == 0) CNT_OK++;
      else {
        CNT_OK--;
      }
    } while (CNT_OK != n);
    for (auto t : qs) {
      cout << TTT[t % cnt] + 1 << ' ';
    }
  }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int tests = 1;
    // cin >> tests;
    while (tests--) {
        solve();
    }
    return 0;
}
```

Task F

```
//                  (     ):  24fin0359

//       :  offecreen // #pragma GCC optimize("O3")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,tune=
    native")

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long

// DATA TYPES
using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ll = long long;
//using ull = unsigned ll;
using ld = long double;
```

```
// DEFINES
#define EACH(el, v) for (auto &el : v)
#define SZ(x) (int)x.size()
#define ALL(x) x.begin(), x.end()

// DEBUG
#define debug(x) cout << #x << " = " << x << endl;

#ifdef UWU
mt19937 rnd(228);
#else
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count
    ());
#endif

const int N = 55;

int T[N][N];
bool TAKEN[N][N];
char RES[N][N];

bool comp(pair<int, int> p1, pair<int, int> p2) {
  return T[p1.first][p1.second] > T[p2.first][p2.second];
}

bool comp2(pair<pair<int, int>, char> p1, pair<pair<int, int>, char> p2)
    {
  return T[p1.first.first][p1.first.second] > T[p2.first.first][p2.first.
    second];
}

int h, w;

int modh(int x) {
  if (x >= h) return x - h;
  if (x < 0) return x + h;
  return x;
}

int modw(int x) {
  if (x >= w) return x - w;
  if (x < 0) return x + w;
  return x;
}

vector<pair<pair<int, int>, char>> SDV = {
  {{-1, 0}, 'T'},
  {{0, 1}, 'R'},
  {{1, 0}, 'B'},
  {{0, -1}, 'L'}
};
```

```
void solve() {
  cin >> h >> w;
  for (int i = 0; i < h; ++i) {
    for (int j = 0; j < w; ++j) {
      RES[i][j] = '.';
    }
  }
  vector<pair<int, int>> idxs;
  for (int i = 0; i < h; ++i) {
    for (int j = 0; j < w; ++j) {
      char c;
      cin >> c;
      T[i][j] = c - '0';
      idxs.push_back({i, j});
    }
  }
  sort(ALL(idxs), comp);
  int ANS = 0;
  for (auto [x, y] : idxs) {
    if (TAKEN[x][y]) continue;
    vector<pair<pair<int, int>, char>> pos;
    for (auto [s, ch] : SDV) {
      int to_x = modh(x + s.first);
      int to_y = modw(y + s.second);
      if (TAKEN[to_x][to_y]) continue;
      pos.push_back({{to_x, to_y}, ch});
    }
    if (pos.empty()) continue;
    sort(ALL(pos), comp2);
    ANS += T[x][y] * T[pos[0].first.first][pos[0].first.second];
    TAKEN[x][y] = true;
    TAKEN[pos[0].first.first][pos[0].first.second] = true;
    if (pos[0].second == 'T') {
      RES[pos[0].first.first][pos[0].first.second] = 'T';
      RES[x][y] = 'B';
    }
    if (pos[0].second == 'R') {
      RES[pos[0].first.first][pos[0].first.second] = 'R';
      RES[x][y] = 'L';
    }
    if (pos[0].second == 'B') {
      RES[pos[0].first.first][pos[0].first.second] = 'B';
      RES[x][y] = 'T';
    }
    if (pos[0].second == 'L') {
      RES[pos[0].first.first][pos[0].first.second] = 'L';
      RES[x][y] = 'R';
    }
  }
  cout << ANS << '\n';
  for (int i = 0; i < h; ++i) {
    for (int j = 0; j < w; ++j) {
      cout << RES[i][j];
```

```
    }
    cout << '\n';
  }
}

int32_t main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);

#ifdef UWU
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int tests = 1;
    // cin >> tests;
    while (tests--) {
        solve();
    }
    return 0;
}
```