Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0339 – Ольхов Иван Дмитриевич

A	B	C	D	E	F	sum
100	100	100	75	20	46	441

Task A

```
/*
&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10368;&#10240;&#
&#10240;&#10240;&#10244;&#10240;&#10244;&#10240;&#10244;&#10256;&#10248;&#10240;&#10272;&#
&#10256;&#10304;&#10244;&#10272;&#10240;&#10244;&#10272;&#10240;&#10244;&#10305;&#10370;&#
&#10384;&#10384;&#10280;&#10280;&#10280;&#10312;&#10306;&#10245;&#10306;&#10370;&#10370;&#
&#10242;&#10274;&#10280;&#10378;&#10252;&#10306;&#10346;&#10384;&#10280;&#10320;&#10388;&#
&#10241;&#10245;&#10261;&#10337;&#10280;&#10306;&#10314;&#10260;&#10344;&#10344;&#10338;&#
&#10280;&#10280;&#10344;&#10384;&#10385;&#10252;&#10338;&#10385;&#10260;&#10396;&#10284;&#
&#10280;&#10280;&#10320;&#10273;&#10306;&#10389;&#10280;&#10306;&#10261;&#10325;&#10325;&#
&#10245;&#10389;&#10280;&#10280;&#10306;&#10389;&#10280;&#10320;&#10389;&#10389;&#10389;&#
&#10245;&#10373;&#10282;&#10280;&#10306;&#10338;&#10321;&#10380;&#10410;&#10410;&#10403;&#
&#10314;&#10260;&#10344;&#10250;&#10388;&#10384;&#10252;&#10274;&#10353;&#10353;&#10467;&#
&#10336;&#10385;&#10244;&#10307;&#10373;&#10274;&#10273;&#10321;&#10396;&#10396;&#10414;&#
&#10318;&#10310;&#10439;&#10382;&#10382;&#10382;&#10325;&#10460;&#10396;&#10333;&#10364;&#
&#10410;&#10378;&#10282;&#10314;&#10325;&#10389;&#10261;&#10389;&#10389;&#10425;&#10296;&#
```

⡧⡣⣏⢮⡪⣎⢮⢣⣣⡣⣇&#
⡝⡎⡞⣜⢮⡚⡮⡳⣕⢳⢕&#
⢪⢌⢪⢪⢪⢪⢸⢨⢢⢣⢣&#
⡳⡱⡱⡱⡣⡣⣣⢳⢕⢧⢳&#
⣕⢕⢕⢝⡜⣎⢮⡣⣏⢮⡣&#
⡧⡣⡣⡳⡕⡵⡱⣕⡕⡧⡳&#
⣕⢇⢇⢏⢮⢪⡣⡧⡳⣹⢪&#
⣎⢎⢎⢗⡕⣇⢗⢵⢝⢎⢧&#
⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#
⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#
⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#
⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#
⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#
⡕⡕⣕⢗⡕⡕⢕⠑⢌⢂⢂&#
⢜⢜⢜⢵⢱⢨⠢⡑⢔⡐⢔&#
⡇⡇⣗⢽⢕⢵⢝⢮⢣⡳⣣&#
⢕⢘⢜⢎⢇⠣⡑⡕⡑⢅⠣&#
⡪⡸⡸⡕⡕⡡⡃⡎⢌⠂⠌&#
⡪⢸⢸⢪⢒⢐⠕⡜⡌⡌⠌&#
⡸⡘⣜⢕⠥⡡⢱⢑⢕⠠⡁&#
⡪⡸⡸⡪⡊⡔⢕⡑⡕⢜⠬&#
⢕⠜⡜⡬⠪⡊⠎⢌⠌⡂⠌&#
⢦⡻⣪⠪⠨⠨⢊⢐⠐⠠⠨&#
⣗⢽⢸⠈⢌⠨⢐⠐⡈⡐⠠&#
⡮⡪⡪⡊⡀⢂⠂⡂⢂⢐⠈&#
⢜⡼⣜⣖⢔⠕⡜⠜⠜⠔⠍&#
⢮⢻⠱⡑⡁⢐⠠⠁⠅⠡⠈&#

```
&#10384;&#10245;&#10339;&#10321;&#10272;&#10240;&#10370;&#10273;&#10280;&#10272;&#10241;&#

&#10256;&#10353;&#10280;&#10328;&#10252;&#10256;&#10304;&#10370;&#10242;&#10252;&#10304;&#

&#10256;&#10380;&#10378;&#10380;&#10370;&#10273;&#10256;&#10320;&#10280;&#10272;&#10256;&#

&#10248;&#10320;&#10373;&#10274;&#10307;&#10244;&#10245;&#10306;&#10245;&#10245;&#10242;&#

&#10244;&#10312;&#10374;&#10289;&#10321;&#10312;&#10384;&#10272;&#10273;&#10312;&#10244;&#

&#10314;&#10244;&#10274;&#10385;&#10388;&#10384;&#10260;&#10337;&#10371;&#10282;&#10312;&#

&#10240;&#10312;&#10296;&#10360;&#10353;&#10337;&#10250;&#10388;&#10273;&#10378;&#10384;&#

&#10242;&#10304;&#10280;&#10328;&#10332;&#10308;&#10245;&#10261;&#10280;&#10384;&#10256;&#

&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#

*/

// #pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")

#include <bits/stdc++.h>
#include <bits/extc++.h>

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
```

```
const int mod2 = 1234567891;
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;

mt19937 rnd(time(0));

void solve() {
    int a, b, n;
    cin >> a >> b >> n;

    int ans = 0;

    if (a <= n) {
        for (int start = 0; start < a; ++start) {
            vector<int> dp(n + 1, 0);
            dp[start] = 1;
            for (int i = start + 1; i <= n - (a - start); ++i) {
                if (i - a >= 0)
                    dp[i] += dp[i - a];
                if (i - b >= 0)
                    dp[i] += dp[i - b];
            }
            ans += dp[n - (a - start)];
        }
    }

    if (b <= n) {
        for (int start = 0; start < b; ++start) {
            vector<int> dp(n + 1, 0);
            dp[start] = 1;
            for (int i = start + 1; i <= n - (b - start); ++i) {
                if (i - a >= 0)
                    dp[i] += dp[i - a];
                if (i - b >= 0)
                    dp[i] += dp[i - b];
            }
            ans += dp[n - (b - start)];
        }
    }

    cout << ans << "\n";

}

signed main() {
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
```

```
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task B

```
/*
&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10368;&#10240;&#

&#10240;&#10240;&#10244;&#10240;&#10244;&#10240;&#10244;&#10256;&#10248;&#10240;&#10272;&#

&#10256;&#10304;&#10244;&#10272;&#10240;&#10244;&#10272;&#10240;&#10244;&#10305;&#10370;&#

&#10384;&#10384;&#10280;&#10280;&#10280;&#10312;&#10306;&#10245;&#10306;&#10370;&#10370;&#

&#10242;&#10274;&#10280;&#10378;&#10252;&#10306;&#10346;&#10384;&#10280;&#10320;&#10388;&#

&#10241;&#10245;&#10261;&#10337;&#10280;&#10306;&#10314;&#10260;&#10344;&#10344;&#10338;&#

&#10280;&#10280;&#10344;&#10384;&#10385;&#10252;&#10338;&#10385;&#10260;&#10396;&#10284;&#

&#10280;&#10280;&#10320;&#10273;&#10306;&#10389;&#10280;&#10306;&#10261;&#10325;&#10325;&#

&#10245;&#10389;&#10280;&#10280;&#10306;&#10389;&#10280;&#10320;&#10389;&#10389;&#10389;&#

&#10245;&#10373;&#10282;&#10280;&#10306;&#10338;&#10321;&#10380;&#10410;&#10410;&#10403;&#

&#10314;&#10260;&#10344;&#10250;&#10388;&#10384;&#10252;&#10274;&#10353;&#10353;&#10467;&#

&#10336;&#10385;&#10244;&#10307;&#10373;&#10274;&#10273;&#10321;&#10396;&#10396;&#10414;&#

&#10318;&#10310;&#10439;&#10382;&#10382;&#10382;&#10325;&#10460;&#10396;&#10333;&#10364;&#

&#10410;&#10378;&#10282;&#10314;&#10325;&#10389;&#10261;&#10389;&#10389;&#10425;&#10296;&#

&#10343;&#10339;&#10447;&#10414;&#10346;&#10446;&#10414;&#10403;&#10467;&#10339;&#10439;&#

&#10333;&#10318;&#10334;&#10460;&#10414;&#10330;&#10350;&#10355;&#10453;&#10419;&#10389;&#

&#10410;&#10380;&#10410;&#10410;&#10410;&#10410;&#10424;&#10408;&#10402;&#10403;&#10403;&#

&#10355;&#10353;&#10353;&#10353;&#10339;&#10339;&#10467;&#10419;&#10389;&#10407;&#10419;&#

&#10453;&#10389;&#10389;&#10397;&#10332;&#10446;&#10414;&#10339;&#10447;&#10414;&#10339;&#

&#10343;&#10339;&#10339;&#10355;&#10325;&#10357;&#10353;&#10453;&#10325;&#10343;&#10355;&#

&#10453;&#10375;&#10375;&#10383;&#10414;&#10410;&#10339;&#10343;&#10355;&#10489;&#10410;&#

&#10446;&#10382;&#10382;&#10391;&#10325;&#10439;&#10391;&#10421;&#10397;&#10382;&#10407;&#
```

⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#

⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#

⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#

⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#

⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#

⡕⡕⣕⢗⡕⡕⢕⠑⢌⢂⢂&#

⢜⢜⢜⢵⢱⢨⠢⡑⢔⡐⢔&#

⡇⡇⣗⢽⢕⢵⢝⢮⢣⡳⣣&#

⢕⢘⢜⢎⢇⠣⡑⡕⡑⢅⠣&#

⡪⡸⡸⡕⡕⡡⡃⡎⢌⠂⠌&#

⡪⢸⢸⢪⢒⢐⠕⡜⡌⡌⠌&#

⡸⡘⣜⢕⠥⡡⢱⢑⢕⠠⡁&#

⡪⡸⡸⡪⡊⡔⢕⡑⡕⢜⠬&#

⢕⠜⡜⡬⠪⡊⠎⢌⠌⡂⠌&#

⢦⡻⣪⠪⠨⠨⢊⢐⠐⠠⠨&#

⣗⢽⢸⠈⢌⠨⢐⠐⡈⡐⠠&#

⡮⡪⡪⡊⡀⢂⠂⡂⢂⢐⠈&#

⢜⡼⣜⣖⢔⠕⡜⠜⠜⠔⠍&#

⢮⢻⠱⡑⡁⢐⠠⠁⠅⠡⠈&#

⢐⠅⡣⡑⠠⠀⢂⠡⠨⠠⠁&#

⠐⡱⠨⡘⠌⠐⡀⢂⠂⠌⡀&#

⠐⢌⢊⢌⢂⠡⠐⡐⠨⠠⠐&#

⠈⡐⢅⠢⡃⠄⠅⡂⠅⠅⠂&#

⠄⡈⢆⠱⡑⡈⢐⠠⠡⡈⠄&#

⡊⠄⠢⢑⢔⢐⠔⡡⢃⠪⡈&#

⠀⡈⠸⡸⡱⡡⠊⢔⠡⢊⢐&#

⠂⡀⠨⡘⡜⡄⠅⠕⠨⢐⠐&#

```
&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#

*/

// #pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")

#include <bits/stdc++.h>
#include <bits/extc++.h>

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
const int mod2 = 1234567891;
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;

mt19937 rnd(time(0));

int get_dist(int n, int x1, int y1, int z1, int x2, int y2, int z2) {
    int dx = abs(x1 - x2), dy = abs(y1 - y2), dz = abs(z1 - z2);

    int mx = max({ dx, dy, dz });
    int mn = min({ dx, dy, dz });
    int mid = dx + dy + dz - mx - mn;

    int sum = mx + mn + mid;
```

```
    sum -= 2 * mx;
    if (sum <= 0) {
        return mx;
    }

    int ans = mx;

    int d = mid - mn;
    mx -= d;
    mid = mn - (mx / 2 + mx % 2);

    ans += mid;

    return ans;
}

void solve() {
    int n;
    cin >> n;

    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;

    int x2, y2, z2;
    cin >> x2 >> y2 >> z2;

    int ans = get_dist(n, x1, y1, z1, x2, y2, z2);
    ans = min(ans, get_dist(n, x1, y1, z1, x2 + 2 * n, y2 - n, z2 - n));
    ans = min(ans, get_dist(n, x1, y1, z1, x2 - 2 * n, y2 + n, z2 + n));
    ans = min(ans, get_dist(n, x1, y1, z1, x2 - n, y2 + 2 * n, z2 - n));
    ans = min(ans, get_dist(n, x1, y1, z1, x2 + n, y2 - 2 * n, z2 + n));
    ans = min(ans, get_dist(n, x1, y1, z1, x2 - n, y2 - n, z2 + 2 * n));
    ans = min(ans, get_dist(n, x1, y1, z1, x2 + n, y2 + n, z2 - 2 * n));

    cout << ans << "\n";
}

signed main() {
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task C

```
/*
```

⠀⠀⠀⠀⠀⠀⠀⠀⠀⢀⠀&#

⠀⠀⠄⠀⠄⠀⠄⠐⠈⠀⠠&#

⠐⡀⠄⠠⠀⠄⠠⠀⠄⡁⢂&#

⢐⢐⠨⠨⠨⡈⡂⠅⡂⢂⢂&#

⠂⠢⠨⢊⠌⡂⡪⢐⠨⡐⢔&#

⠁⠅⠕⡡⠨⡂⡊⠔⡨⡨⡢&#

⠨⠨⡨⢐⢑⠌⡢⢑⠔⢜⠬&#

⠨⠨⡐⠡⡂⢕⠨⡂⠕⡕⡕&#

⠅⢕⠨⠨⡂⢕⠨⡐⢕⢕⢕&#

⠅⢅⠪⠨⡂⡢⡑⢌⢪⢪⢣&#

⡊⠔⡨⠊⢔⢐⠌⠢⡱⡱⣣&#

⡠⢑⠄⡃⢅⠢⠡⡑⢜⢜⢮&#

⡎⡆⣇⢎⢎⢎⡕⣜⢜⡝⡼&#

⢪⢊⠪⡊⡕⢕⠕⢕⢕⢹⠸&#

⡧⡣⣏⢮⡪⣎⢮⢣⣣⡣⣇&#

⡝⡎⡞⣜⢮⡚⡮⡳⣕⢳⢕&#

⢪⢌⢪⢪⢪⢪⢸⢨⢢⢣⢣&#

⡳⡱⡱⡱⡣⡣⣣⢳⢕⢧⢳&#

⣕⢕⢕⢝⡜⣎⢮⡣⣏⢮⡣&#

⡧⡣⡣⡳⡕⡵⡱⣕⡕⡧⡳&#

⣕⢇⢇⢏⢮⢪⡣⡧⡳⣹⢪&#

⣎⢎⢎⢗⡕⣇⢗⢵⢝⢎⢧&#

⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#

⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#

⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#

⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#

⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#

```
&#10325;&#10325;&#10453;&#10391;&#10325;&#10325;&#10389;&#10257;&#10380;&#10370;&#10370;&#
&#10396;&#10396;&#10396;&#10421;&#10417;&#10408;&#10274;&#10321;&#10388;&#10320;&#10388;&#
&#10311;&#10311;&#10455;&#10429;&#10389;&#10421;&#10397;&#10414;&#10403;&#10355;&#10467;&#
&#10389;&#10392;&#10396;&#10382;&#10375;&#10275;&#10321;&#10325;&#10321;&#10373;&#10275;&#
&#10346;&#10360;&#10360;&#10325;&#10325;&#10337;&#10307;&#10318;&#10380;&#10242;&#10252;&#
&#10346;&#10424;&#10424;&#10410;&#10386;&#10384;&#10261;&#10332;&#10316;&#10316;&#10252;&#
&#10360;&#10328;&#10460;&#10389;&#10277;&#10337;&#10417;&#10385;&#10389;&#10272;&#10305;&#
&#10346;&#10360;&#10360;&#10346;&#10314;&#10324;&#10389;&#10321;&#10325;&#10396;&#10284;&#
&#10389;&#10268;&#10332;&#10348;&#10282;&#10314;&#10254;&#10380;&#10252;&#10306;&#10252;&#
&#10406;&#10363;&#10474;&#10282;&#10280;&#10280;&#10378;&#10384;&#10256;&#10272;&#10280;&#
&#10455;&#10429;&#10424;&#10248;&#10380;&#10280;&#10384;&#10256;&#10312;&#10320;&#10272;&#
&#10350;&#10346;&#10346;&#10314;&#10304;&#10370;&#10242;&#10306;&#10370;&#10384;&#10248;&#
&#10396;&#10364;&#10460;&#10454;&#10388;&#10261;&#10332;&#10268;&#10268;&#10260;&#10253;&#
&#10414;&#10427;&#10289;&#10321;&#10305;&#10384;&#10272;&#10241;&#10245;&#10273;&#10248;&#
&#10384;&#10245;&#10339;&#10321;&#10272;&#10240;&#10370;&#10273;&#10280;&#10272;&#10241;&#
&#10256;&#10353;&#10280;&#10328;&#10252;&#10256;&#10304;&#10370;&#10242;&#10252;&#10304;&#
&#10256;&#10380;&#10378;&#10380;&#10370;&#10273;&#10256;&#10320;&#10280;&#10272;&#10256;&#
&#10248;&#10320;&#10373;&#10274;&#10307;&#10244;&#10245;&#10306;&#10245;&#10245;&#10242;&#
&#10244;&#10312;&#10374;&#10289;&#10321;&#10312;&#10384;&#10272;&#10273;&#10312;&#10244;&#
&#10314;&#10244;&#10274;&#10385;&#10388;&#10384;&#10260;&#10337;&#10371;&#10282;&#10312;&#
&#10240;&#10312;&#10296;&#10360;&#10353;&#10337;&#10250;&#10388;&#10273;&#10378;&#10384;&#
&#10242;&#10304;&#10280;&#10328;&#10332;&#10308;&#10245;&#10261;&#10280;&#10384;&#10256;&#
&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#
*/

// #pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")
```

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

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
const int mod2 = 1234567891;
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;

mt19937 rnd(time(0));

int sub(int a, int b) {
    if (a - b < 0)
        return a - b + M;
    return a - b;
}

int add(int a, int b) {
    if (a + b >= M)
        return a + b - M;
    return a + b;
}

struct hash_string {
    vector<int> hash;
    int p = 31;
    vector<int> powers = { 1 };
    int power(int x) {
```

```
        while (x >= (int)powers.size())
            powers.push_back(1ll * p * powers.back() % M);
        return powers[x];
    }

    hash_string(string s) {
        int n = s.size();
        hash.resize(n + 1, 0);
        for (int i = 1; i <= n; ++i) {
            hash[i] = add(1ll * hash[i - 1] * p % M,  s[i - 1]);
        }
    }

    int get_hash(int l, int r) {
        return sub(hash[r], 1ll * hash[l] * power(r - l) % M);
    }
};

void solve() {
    string command;
    cin >> command;
    if (command == "first") {
        int n;
        cin >> n;
        string s;
        cin >> s;

        hash_string hs(s);
        cout << hs.get_hash(0, n);

    } else {
        int n;
        cin >> n;
        string t;
        cin >> t;

        int hs;
        cin >> hs;
        hash_string nhs(t + t);

        for (int i = n; i <= 2 * n; ++i) {
            int cur_hash = nhs.get_hash(i - n, i);
            if (hs == cur_hash) {
                cout << "YES\n";
                return;
            }
        }

        cout << "NO\n";
    }
}

signed main() {
```

```
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task D

```
/*
&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10368;&#10240;&#

&#10240;&#10240;&#10244;&#10240;&#10244;&#10240;&#10244;&#10256;&#10248;&#10240;&#10272;&#

&#10256;&#10304;&#10244;&#10272;&#10240;&#10244;&#10272;&#10240;&#10244;&#10305;&#10370;&#

&#10384;&#10384;&#10280;&#10280;&#10280;&#10312;&#10306;&#10245;&#10306;&#10370;&#10370;&#

&#10242;&#10274;&#10280;&#10378;&#10252;&#10306;&#10346;&#10384;&#10280;&#10320;&#10388;&#

&#10241;&#10245;&#10261;&#10337;&#10280;&#10306;&#10314;&#10260;&#10344;&#10344;&#10338;&#

&#10280;&#10280;&#10344;&#10384;&#10385;&#10252;&#10338;&#10385;&#10260;&#10396;&#10284;&#

&#10280;&#10280;&#10320;&#10273;&#10306;&#10389;&#10280;&#10306;&#10261;&#10325;&#10325;&#

&#10245;&#10389;&#10280;&#10280;&#10306;&#10389;&#10280;&#10320;&#10389;&#10389;&#10389;&#

&#10245;&#10373;&#10282;&#10280;&#10306;&#10338;&#10321;&#10380;&#10410;&#10410;&#10403;&#

&#10314;&#10260;&#10344;&#10250;&#10388;&#10384;&#10252;&#10274;&#10353;&#10353;&#10467;&#

&#10336;&#10385;&#10244;&#10307;&#10373;&#10274;&#10273;&#10321;&#10396;&#10396;&#10414;&#

&#10318;&#10310;&#10439;&#10382;&#10382;&#10382;&#10325;&#10460;&#10396;&#10333;&#10364;&#

&#10410;&#10378;&#10282;&#10314;&#10325;&#10389;&#10261;&#10389;&#10389;&#10425;&#10296;&#

&#10343;&#10339;&#10447;&#10414;&#10346;&#10446;&#10414;&#10403;&#10467;&#10339;&#10439;&#

&#10333;&#10318;&#10334;&#10460;&#10414;&#10330;&#10350;&#10355;&#10453;&#10419;&#10389;&#

&#10410;&#10380;&#10410;&#10410;&#10410;&#10410;&#10424;&#10408;&#10402;&#10403;&#10403;&#

&#10355;&#10353;&#10353;&#10353;&#10339;&#10339;&#10467;&#10419;&#10389;&#10407;&#10419;&#

&#10453;&#10389;&#10389;&#10397;&#10332;&#10446;&#10414;&#10339;&#10447;&#10414;&#10339;&#
```

⡧⡣⡣⡳⡕⡵⡱⣕⡕⡧⡳&#
⣕⢇⢇⢏⢮⢪⡣⡧⡳⣹⢪&#
⣎⢎⢎⢗⡕⣇⢗⢵⢝⢎⢧&#
⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#
⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#
⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#
⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#
⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#
⡕⡕⣕⢗⡕⡕⢕⠑⢌⢂⢂&#
⢜⢜⢜⢵⢱⢨⠢⡑⢔⡐⢔&#
⡇⡇⣗⢽⢕⢵⢝⢮⢣⡳⣣&#
⢕⢘⢜⢎⢇⠣⡑⡕⡑⢅⠣&#
⡪⡸⡸⡕⡕⡡⡃⡎⢌⠂⠌&#
⡪⢸⢸⢪⢒⢐⠕⡜⡌⡌⠌&#
⡸⡘⣜⢕⠥⡡⢱⢑⢕⠠⡁&#
⡪⡸⡸⡪⡊⡔⢕⡑⡕⢜⠬&#
⢕⠜⡜⡬⠪⡊⠎⢌⠌⡂⠌&#
⢦⡻⣪⠪⠨⠨⢊⢐⠐⠠⠨&#
⣗⢽⢸⠈⢌⠨⢐⠐⡈⡐⠠&#
⡮⡪⡪⡊⡀⢂⠂⡂⢂⢐⠈&#
⢜⡼⣜⣖⢔⠕⡜⠜⠜⠔⠍&#
⢮⢻⠱⡑⡁⢐⠠⠁⠅⠡⠈&#
⢐⠅⡣⡑⠠⠀⢂⠡⠨⠠⠁&#
⠐⡱⠨⡘⠌⠐⡀⢂⠂⠌⡀&#
⠐⢌⢊⢌⢂⠡⠐⡐⠨⠠⠐&#
⠈⡐⢅⠢⡃⠄⠅⡂⠅⠅⠂&#
⠄⡈⢆⠱⡑⡈⢐⠠⠡⡈⠄&#

```
&#10314;&#10244;&#10274;&#10385;&#10388;&#10384;&#10260;&#10337;&#10371;&#10282;&#10312;&#

&#10240;&#10312;&#10296;&#10360;&#10353;&#10337;&#10250;&#10388;&#10273;&#10378;&#10384;&#

&#10242;&#10304;&#10280;&#10328;&#10332;&#10308;&#10245;&#10261;&#10280;&#10384;&#10256;&#

&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#

*/

#pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")

#include <bits/stdc++.h>
#include <bits/extc++.h>

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
const int mod2 = 1234567891;
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;

mt19937 rnd(time(0));

char ask(int i, int j) {
    cout << i + 1 << " " << j + 1 << endl;
```

```
    char res;
    cin >> res;
    if (res == '!')
        exit(0);
    return res;
}

void solve() {
    int n;
    cin >> n;

    vector<char> p(n);
    for (int i = 0; i < n; ++i) {
        char res = ask(0, 0);
        if (res == '<')
            p[i] = '>';
        else
            p[i] = '<';
    }


    vector<char> idol(n, '>');

    while (p != idol) {
        int id = -1;
        for (int i = n - 1; i >= 0; --i) {
            if (p[i] == '<') {
                id = i;
                break;
            }
        }

        ask(id, (id + 1) % n);
        for (int i = 1; i < n; ++i) {
            ask(0, 0);
        }

        p.clear();
        for (int i = 0; i < n; ++i) {
            char res = ask(0, 0);
            if (res == '<')
                p[i] = '>';
            else
                p[i] = '<';
        }
    }

    for (int i = 0; i < n; ++i) {
        char res = ask(0, 0);
        if (res == '<')
            p[i] = '>';
        else
            p[i] = '<';
```

```
    }
}

signed main() {
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task E

```
/*
&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10368;&#10240;&#

&#10240;&#10240;&#10244;&#10240;&#10244;&#10240;&#10244;&#10256;&#10248;&#10240;&#10272;&#

&#10256;&#10304;&#10244;&#10272;&#10240;&#10244;&#10272;&#10240;&#10244;&#10305;&#10370;&#

&#10384;&#10384;&#10280;&#10280;&#10280;&#10312;&#10306;&#10245;&#10306;&#10370;&#10370;&#

&#10242;&#10274;&#10280;&#10378;&#10252;&#10306;&#10346;&#10384;&#10280;&#10320;&#10388;&#

&#10241;&#10245;&#10261;&#10337;&#10280;&#10306;&#10314;&#10260;&#10344;&#10344;&#10338;&#

&#10280;&#10280;&#10344;&#10384;&#10385;&#10252;&#10338;&#10385;&#10260;&#10396;&#10284;&#

&#10280;&#10280;&#10320;&#10273;&#10306;&#10389;&#10280;&#10306;&#10261;&#10325;&#10325;&#

&#10245;&#10389;&#10280;&#10280;&#10306;&#10389;&#10280;&#10320;&#10389;&#10389;&#10389;&#

&#10245;&#10373;&#10282;&#10280;&#10306;&#10338;&#10321;&#10380;&#10410;&#10410;&#10403;&#

&#10314;&#10260;&#10344;&#10250;&#10388;&#10384;&#10252;&#10274;&#10353;&#10353;&#10467;&#

&#10336;&#10385;&#10244;&#10307;&#10373;&#10274;&#10273;&#10321;&#10396;&#10396;&#10414;&#

&#10318;&#10310;&#10439;&#10382;&#10382;&#10382;&#10325;&#10460;&#10396;&#10333;&#10364;&#

&#10410;&#10378;&#10282;&#10314;&#10325;&#10389;&#10261;&#10389;&#10389;&#10425;&#10296;&#

&#10343;&#10339;&#10447;&#10414;&#10346;&#10446;&#10414;&#10403;&#10467;&#10339;&#10439;&#

&#10333;&#10318;&#10334;&#10460;&#10414;&#10330;&#10350;&#10355;&#10453;&#10419;&#10389;&#

&#10410;&#10380;&#10410;&#10410;&#10410;&#10410;&#10424;&#10408;&#10402;&#10403;&#10403;&#
```

⡳⡱⡱⡱⡣⡣⣣⢳⢕⢧⢳&#
⣕⢕⢕⢝⡜⣎⢮⡣⣏⢮⡣&#
⡧⡣⡣⡳⡕⡵⡱⣕⡕⡧⡳&#
⣕⢇⢇⢏⢮⢪⡣⡧⡳⣹⢪&#
⣎⢎⢎⢗⡕⣇⢗⢵⢝⢎⢧&#
⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#
⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#
⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#
⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#
⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#
⡕⡕⣕⢗⡕⡕⢕⠑⢌⢂⢂&#
⢜⢜⢜⢵⢱⢨⠢⡑⢔⡐⢔&#
⡇⡇⣗⢽⢕⢵⢝⢮⢣⡳⣣&#
⢕⢘⢜⢎⢇⠣⡑⡕⡑⢅⠣&#
⡪⡸⡸⡕⡕⡡⡃⡎⢌⠂⠌&#
⡪⢸⢸⢪⢒⢐⠕⡜⡌⡌⠌&#
⡸⡘⣜⢕⠥⡡⢱⢑⢕⠠⡁&#
⡪⡸⡸⡪⡊⡔⢕⡑⡕⢜⠬&#
⢕⠜⡜⡬⠪⡊⠎⢌⠌⡂⠌&#
⢦⡻⣪⠪⠨⠨⢊⢐⠐⠠⠨&#
⣗⢽⢸⠈⢌⠨⢐⠐⡈⡐⠠&#
⡮⡪⡪⡊⡀⢂⠂⡂⢂⢐⠈&#
⢜⡼⣜⣖⢔⠕⡜⠜⠜⠔⠍&#
⢮⢻⠱⡑⡁⢐⠠⠁⠅⠡⠈&#
⢐⠅⡣⡑⠠⠀⢂⠡⠨⠠⠁&#
⠐⡱⠨⡘⠌⠐⡀⢂⠂⠌⡀&#
⠐⢌⢊⢌⢂⠡⠐⡐⠨⠠⠐&#

```
&#10248;&#10320;&#10373;&#10274;&#10307;&#10244;&#10245;&#10306;&#10245;&#10245;&#10242;&#

&#10244;&#10312;&#10374;&#10289;&#10321;&#10312;&#10384;&#10272;&#10273;&#10312;&#10244;&#

&#10314;&#10244;&#10274;&#10385;&#10388;&#10384;&#10260;&#10337;&#10371;&#10282;&#10312;&#

&#10240;&#10312;&#10296;&#10360;&#10353;&#10337;&#10250;&#10388;&#10273;&#10378;&#10384;&#

&#10242;&#10304;&#10280;&#10328;&#10332;&#10308;&#10245;&#10261;&#10280;&#10384;&#10256;&#

&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#

*/

// #pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")

#include <bits/stdc++.h>
#include <bits/extc++.h>

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
const int mod2 = 1234567891;
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;
```

```
mt19937 rnd(time(0));

vector<queue<int>> g;
vector<int> starts;
vector<int> time_points;

bool check() {
    for (int i = 0; i < (int)starts.size(); ++i) {
        if (starts[i] != g[i].front())
            return false;
    }
    return true;
}

int timer = 0;
void dfs(int v) {
    time_points.push_back(v);
    if (timer == 0) return;

    // if (check()) {
    //     return;
    // } else if (starts[v] == -1) {
    //     starts[v] = g[v].front();
    // }
    --timer;
    g[v].push(g[v].front());
    g[v].pop();
    dfs(g[v].back());
}

void solve() {
    int n, q;
    cin >> n >> q;

    g.resize(n);
    starts.assign(n, -1);
    for (int i = 0; i < n; ++i) {
        int m;
        cin >> m;
        while (m--) {
            int x;
            cin >> x;
            --x;
            g[i].push(x);
        }
    }


    vector<int> queries(q);
    for (auto& el : queries)
        cin >> el;
    timer = *max_element(all(queries));
```

```
    dfs(0);
    for (auto& el : queries) {
        cout << time_points[el] + 1 << " ";
    }

    cout << "\n";
}

signed main() {
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
}
```

Task F

```
/*
&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10240;&#10368;&#10240;&#

&#10240;&#10240;&#10244;&#10240;&#10244;&#10240;&#10244;&#10256;&#10248;&#10240;&#10272;&#

&#10256;&#10304;&#10244;&#10272;&#10240;&#10244;&#10272;&#10240;&#10244;&#10305;&#10370;&#

&#10384;&#10384;&#10280;&#10280;&#10280;&#10312;&#10306;&#10245;&#10306;&#10370;&#10370;&#

&#10242;&#10274;&#10280;&#10378;&#10252;&#10306;&#10346;&#10384;&#10280;&#10320;&#10388;&#

&#10241;&#10245;&#10261;&#10337;&#10280;&#10306;&#10314;&#10260;&#10344;&#10344;&#10338;&#

&#10280;&#10280;&#10344;&#10384;&#10385;&#10252;&#10338;&#10385;&#10260;&#10396;&#10284;&#

&#10280;&#10280;&#10320;&#10273;&#10306;&#10389;&#10280;&#10306;&#10261;&#10325;&#10325;&#

&#10245;&#10389;&#10280;&#10280;&#10306;&#10389;&#10280;&#10320;&#10389;&#10389;&#10389;&#

&#10245;&#10373;&#10282;&#10280;&#10306;&#10338;&#10321;&#10380;&#10410;&#10410;&#10403;&#

&#10314;&#10260;&#10344;&#10250;&#10388;&#10384;&#10252;&#10274;&#10353;&#10353;&#10467;&#

&#10336;&#10385;&#10244;&#10307;&#10373;&#10274;&#10273;&#10321;&#10396;&#10396;&#10414;&#

&#10318;&#10310;&#10439;&#10382;&#10382;&#10382;&#10325;&#10460;&#10396;&#10333;&#10364;&#

&#10410;&#10378;&#10282;&#10314;&#10325;&#10389;&#10261;&#10389;&#10389;&#10425;&#10296;&#

&#10343;&#10339;&#10447;&#10414;&#10346;&#10446;&#10414;&#10403;&#10467;&#10339;&#10439;&#
```

⡝⡎⡞⣜⢮⡚⡮⡳⣕⢳⢕&#
⢪⢌⢪⢪⢪⢪⢸⢨⢢⢣⢣&#
⡳⡱⡱⡱⡣⡣⣣⢳⢕⢧⢳&#
⣕⢕⢕⢝⡜⣎⢮⡣⣏⢮⡣&#
⡧⡣⡣⡳⡕⡵⡱⣕⡕⡧⡳&#
⣕⢇⢇⢏⢮⢪⡣⡧⡳⣹⢪&#
⣎⢎⢎⢗⡕⣇⢗⢵⢝⢎⢧&#
⢱⢕⢕⢧⢳⢱⡹⡜⣎⢯⢺&#
⢕⢇⢗⢝⢎⢇⢗⡝⣜⢮⢳&#
⡳⡱⡱⣝⢕⡝⣜⢮⢺⢪⡳&#
⣝⢜⢕⢧⢳⢹⡸⣪⢣⡳⡱&#
⢮⢪⢳⡹⡕⡇⡗⡕⡕⢕⢑&#
⡕⡕⣕⢗⡕⡕⢕⠑⢌⢂⢂&#
⢜⢜⢜⢵⢱⢨⠢⡑⢔⡐⢔&#
⡇⡇⣗⢽⢕⢵⢝⢮⢣⡳⣣&#
⢕⢘⢜⢎⢇⠣⡑⡕⡑⢅⠣&#
⡪⡸⡸⡕⡕⡡⡃⡎⢌⠂⠌&#
⡪⢸⢸⢪⢒⢐⠕⡜⡌⡌⠌&#
⡸⡘⣜⢕⠥⡡⢱⢑⢕⠠⡁&#
⡪⡸⡸⡪⡊⡔⢕⡑⡕⢜⠬&#
⢕⠜⡜⡬⠪⡊⠎⢌⠌⡂⠌&#
⢦⡻⣪⠪⠨⠨⢊⢐⠐⠠⠨&#
⣗⢽⢸⠈⢌⠨⢐⠐⡈⡐⠠&#
⡮⡪⡪⡊⡀⢂⠂⡂⢂⢐⠈&#
⢜⡼⣜⣖⢔⠕⡜⠜⠜⠔⠍&#
⢮⢻⠱⡑⡁⢐⠠⠁⠅⠡⠈&#

```
&#10384;&#10245;&#10339;&#10321;&#10272;&#10240;&#10370;&#10273;&#10280;&#10272;&#10241;&#

&#10256;&#10353;&#10280;&#10328;&#10252;&#10256;&#10304;&#10370;&#10242;&#10252;&#10304;&#

&#10256;&#10380;&#10378;&#10380;&#10370;&#10273;&#10256;&#10320;&#10280;&#10272;&#10256;&#

&#10248;&#10320;&#10373;&#10274;&#10307;&#10244;&#10245;&#10306;&#10245;&#10245;&#10242;&#

&#10244;&#10312;&#10374;&#10289;&#10321;&#10312;&#10384;&#10272;&#10273;&#10312;&#10244;&#

&#10314;&#10244;&#10274;&#10385;&#10388;&#10384;&#10260;&#10337;&#10371;&#10282;&#10312;&#

&#10240;&#10312;&#10296;&#10360;&#10353;&#10337;&#10250;&#10388;&#10273;&#10378;&#10384;&#

&#10242;&#10304;&#10280;&#10328;&#10332;&#10308;&#10245;&#10261;&#10280;&#10384;&#10256;&#

&#10242;&#10240;&#10272;&#10264;&#10332;&#10244;&#10369;&#10241;&#10305;&#10272;&#10256;&#

*/

// #pragma GCC optimize("O3,unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")
// #pragma GCC optimize("Ofast")
// #pragma GCC target("sse4.2")

#include <bits/stdc++.h>
#include <bits/extc++.h>

#define int long long
#define all(c) c.begin(), c.end()

using namespace std;
using namespace __gnu_pbds;

using ll = int64_t;
using ull = uint64_t;
using ld = long double;
using pii = pair<int, int>;
using pll = pair<ll, ll>;
using pdd = pair<double, double>;

using ordered_set = tree<int, null_type, less<int>, rb_tree_tag,
    tree_order_statistics_node_update>;
using ordered_multiset = tree<int, null_type, less_equal<int>,
    rb_tree_tag, tree_order_statistics_node_update>;

///
const int INF = static_cast<int>(1e9) + static_cast<int>(1e3);
const ll INFL = static_cast<ll>(1e18) + static_cast<ll>(1e6);
const double pi = atan2(0., -1.);
const int M = static_cast<int>(1e9) + 7;
const int mod1 = 1337228777;
const int mod2 = 1234567891;
```

```
const int mod3 = 998244353;
const int BLOCK = 750;
const int ITER = 300;
const double TL = 1.;

mt19937 rnd(time(0));

void solve() {
    int h, w;
    cin >> h >> w;
    vector<vector<int>> a(h, vector<int>(w));
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            char c;
            cin >> c;
            a[i][j] = c - '0';
        }
    }

    vector<vector<char>> ans(h, vector<char>(w, '.'));

    int p = 0;
    for (int i = 0; i < h; i += 2) {
        if (i == h - 1) {
            continue;
        }
        for (int j = 0; j < w; ++j) {
            p += a[i][j] * a[i + 1][j];
            ans[i][j] = 'T';
            ans[i + 1][j] = 'B';
        }
    }

    cout << p << "\n";
    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            cout << ans[i][j];
        }
        cout << "\n";
    }
}

signed main() {
    ///
    ios::sync_with_stdio(false);
    cin.tie(nullptr)->sync_with_stdio(false);
    cout.tie(nullptr);

    int t = 1;
    // cin >> t;
    while (t--) solve();

    return 0;
```

```
}
```