Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0196 – Катасонов Роман Алексеевич

A	B	C	D	E	F	sum
100	100	100	35	20	4	359

Task A

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

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

#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
```

```
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
    x.sc; } };

vector<int> dp;

signed main() {
    fast
#ifdef LOCAL
    file_in
#endif

    int a, b, n;
    cin >> a >> b >> n;
    dp.resize(n + 1);
    dp[0] = 1;
    for (int i = 1; i < n; ++i) {
        if (i - a >= 0) dp[i] += dp[i - a];
        if (i - b >= 0) dp[i] += dp[i - b];
    }
    int ans = 0;
    if (n - a >= 0) ans += dp[n - a] * a;
    if (n - b >= 0) ans += dp[n - b] * b;
```

```
    cout << ans;

    return 0;
}
```

Task B

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

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

#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const
```

```
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
    x.sc; } };

int get(int x00, int y00, int z00, int x11, int y11, int z11) {
    int res = 0;

    int X = x00 - x11;
    int Y = y00 - y11;
    int Z = z00 - z11;

    if (X == 0) return abs(Y);
    if (Y == 0 || Z == 0) return abs(X);

    if ((X < 0) == (Y < 0))
        return abs(X) + abs(Y);
    if ((X < 0) == (Z < 0))
        return abs(X) + abs(Z);
    if ((Y < 0) == (Z < 0)) {
        return abs(Y) + abs(Z);
    }

    return 0;
}

int x00, y00, z00;
int get(int x11, int y11, int z11) {
    return get(x00, y00, z00, x11, y11, z11);
}

int res;

int n;
int depth = 5, x11, y11, z11;

struct lol { int x, y, z; };
vector<lol> d = {{2, -1, -1}, {-1, 2, -1}, {-1, -1, 2}, {-2, 1, 1}, {1,
    -2, 1}, {1, 1, -2}};

void rec(int i, int x, int y, int z) {
    res = min(res, get(x11 + x, y11 + y, z11 + z));
```

```
    if (i == depth) return;
    for (auto [d1, d2, d3] : d) {
        rec(i + 1, x + n * d1, y + n * d2, z + n * d3);
    }
}

signed main() {
    fast
#ifdef LOCAL
    file_in
#endif

    cin >> n;
    cin >> x00 >> y00 >> z00;
    cin >> x11 >> y11 >> z11;

    res = get(x00, y00, z00, x11, y11, z11);

    rec(0, 0, 0, 0);

    // res = min(res, get(x00, y00, z00, x11 + 2 * n, y11 - n, z11 - n),
   get(x00, y00, z00, x11 - n, y11 + 2 * n, z11 - n), get(x00, y00, z00,
   x11 - n, y11 - n, z11 + 2 * n));
    // res = min(res, get(x00, y00, z00, x11 - 2 * n, y11 + n, z11 + n),
   get(x00, y00, z00, x11 + n, y11 - 2 * n, z11 + n), get(x00, y00, z00,
   x11 - n, y11 - n, z11 + 2 * n));
    // res = min(res, get(x11 - 3 * n, y11 + n, z11), get(x11 - 3 * n,
   y11, z11 + n), get(x11, y11 - 3 * n, z11 + n));
    // res = min(res, get(x11 + n, y11 - 3 * n, z11), get(x11 + n, y11,
   z11 - 3 * n), get(x11, y11 + n, z11 - 3 * n));
    // res = min(res, get(x11 + 3 * n, y11 - n, z11), get(x11 + 3 * n,
   y11, z11 - n), get(x11, y11 + 3 * n, z11 - n));
    // res = min(res, get(x11 - n, y11 + 3 * n, z11), get(x11 - n, y11,
   z11 + 3 * n), get(x11, y11 - n, z11 + 3 * n));

    cout << res;

    return 0;
}
```

Task C

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

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

#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
```

```
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
    x.sc; } };

const int mod1 = 1e9 + 7, p1 = 59;
const int mod2 = 1e9 + 9, p2 = 257;
```

```
signed main() {
    fast
#ifdef LOCAL
    file_in
#endif

    str type;
    cin >> type;

    if (type == "first") {
        int n;
        cin >> n;
        str s;
        cin >> s;

        int hash1 = 0, hash2 = 0;
        for (auto el : s) {
            hash1 = (hash1 * p1 + el - '0' + 1) % mod1;
            hash2 = (hash2 * p2 + el - '0' + 1) % mod2;
        }
        cout << hash1 << " " << hash2;
    } else {
        int n;
        cin >> n;
        str s;
        cin >> s;
        int hash1 = 0, hash2 = 0;
        for (auto el : s) {
            hash1 = (hash1 * p1 + el - '0' + 1) % mod1;
            hash2 = (hash2 * p2 + el - '0' + 1) % mod2;
        }
        int need1, need2;
        cin >> need1 >> need2;
        if (need1 == hash1 && need2 == hash2) { cout << "YES"; return 0;
    }

        int ppow1 = 1, ppow2 = 1;
        for (int i = 0; i < n - 1; ++i) {
            ppow1 = ppow1 * p1 % mod1;
            ppow2 = ppow2 * p2 % mod2;
        }

        for (int i = 0; i < n; ++i) {
            hash1 -= (s[i] - '0' + 1) * ppow1;
            hash1 = (hash1 % mod1 + mod1) % mod1;
            hash1 = (hash1 * p1 + s[i] - '0' + 1) % mod1;
            hash2 -= (s[i] - '0' + 1) * ppow2;
            hash2 = (hash2 % mod2 + mod2) % mod2;
            hash2 = (hash2 * p2 + s[i] - '0' + 1) % mod2;
            if (need1 == hash1 && need2 == hash2) { cout << "YES"; return
     0; }
        }
```

```
        cout << "NO";
    }

    return 0;
}
```

Task D

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

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

#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
```

```
template<typename T> ostream& operator<<(ostream& out, const
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
    x.sc; } };

char last;

void que(int i, int j) {
    cout << i + 1 << " " << j + 1 << endl;
    cin >> last;
    if (last == '!') exit(0);
}

void wait() { que(0, 0); }
void wait(int n) {
    while (n--)
       wait();
}

int n;
vector<int> p;

void who(int id) {
    for (int i = 0; i < n; ++i) {
        if (i == 0) { // compare p[0] & p[n - 1]
            wait();
            if (last == '<') // p[n - 1] < p[0]
                ++p[id];
        } else { // compare p[0] & p[n - 1 - i]
            que(0, n - i);
            wait(n - 1);

            if (i != n - 1 && last == '<') // p[n - 1 - i] < p[0]
                ++p[id];
        }
    }
}

signed main() {
    fast
#ifdef LOCAL
```

```
    file_in
#endif

    cin >> n;

    if (n <= 5) {
        while (true) {
            int i = randint(1, n), j = randint(1, n);
            cout << i << ' ' << j << endl;
            char cur;
            cin >> cur;
            if (cur == '!')
                return 0;
        }
        return 0;
    }

    p.resize(n, 1);

    for (int i = 0; i < n; ++i) {
        who((n - i) % n);
        wait();
    }

    //        - p

    vector<int> pos(n + 1);

    for (int i = 0; i < n; ++i)
        pos[p[i]] = i;

    for (int i = 0; i < n; ++i) {
        if (i == pos[n - i]) continue;
        que(i, pos[n - i]);
        wait(n - 1);
        swap(pos[p[i]], pos[n - i]);
    }
    wait(2 * n);

    assert(false);

    return 0;
}
```

Task E

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

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

```
#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
```

```
    x.sc; } };

vector<vector<int>> g;
vector<int> cur;

const int inf = 1e5 + 10;

vector<int> ord;

void dfs(int v) {
    while (sz(ord) < inf) {
        ord.pb(v);
        ++cur[v];
        if (cur[v] == sz(g[v])) cur[v] = 0;
        v = g[v][cur[v]];
    }
}

signed main() {
    fast
#ifdef LOCAL
    file_in
#endif

    int n, q;
    cin >> n >> q;

    g.resize(n);
    cur.resize(n);

    for (int i = 0; i < n; ++i) {
        int k;
        cin >> k;
        cur[i] = k - 1;
        g[i].resize(k);
        cin >> g[i];
        for (auto &el : g[i]) --el;
    }

    dfs(0);

    while (q--) {
        int t;
        cin >> t;
        cout << ord[t] + 1 << ' ';
    }

    return 0;
}
```

Task F

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

```
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("avx2,bmi,bmi2,lzcnt,popcnt")

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

#define mt_seed chrono::high_resolution_clock::now().time_since_epoch().
    count()
#define time_now chrono::high_resolution_clock::now()
#define time_res(s, f) chrono::duration_cast<chrono::nanoseconds>(f-s).
    count()
#define fast ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
#define file_in freopen("input.txt", "r", stdin);
#define file_out freopen("output.txt", "w", stdout);
#define str string
#define pb push_back
#define ff first
#define sc second
#define all(a) a.begin(), a.end()
#define sz(a) (int)a.size()
#define pNode pair<Node*, Node*>
#define int ll

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef tree<int, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update> ordered_set;

template<typename T, typename T1> istream& operator>>(istream& in, pair<T
    , T1>& a) { in >> a.ff >> a.sc; return in; }
template<typename T, typename T1> ostream& operator<<(ostream& out, const
     pair<T, T1> &a) { out << a.ff << " " << a.sc; return out; }
template<typename T> istream& operator>>(istream& in, vector<T>& a) { for
     (auto& el : a) in >> el; return in; }
template<typename T> ostream& operator<<(ostream& out, const vector<T>& a
    ) { for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const set<T>& a) {
     for (auto el : a) out << el << " "; return out; }
template<typename T> ostream& operator<<(ostream& out, const
    unordered_set<T>& a) { for (auto el : a) out << el << " "; return out;
     }
template<typename T> T min(T a, T b, T c) { return min(min(a, b), c); }
template<typename T> T max(T a, T b, T c) { return max(max(a, b), c); }
template<typename T> T min(T a, T b, T c, T d) { return min(min(a, b),
    min(c, d)); }
template<typename T> T max(T a, T b, T c, T d) { return max(max(a, b),
    max(c, d)); }
```

```
mt19937_64 genom(mt_seed); ll randint(ll l, ll r) {
    uniform_int_distribution<ll> random_distrib_2_0(l, r); return
    random_distrib_2_0(genom); }
const int RANDOM = mt_seed; struct chash { int operator()(int x) const {
    return x ^ RANDOM; } };
struct pair_hash { ll operator()(pii x) const { return 1ll * x.ff * 31 +
    x.sc; } };

clock_t start = clock();

int h, w;

vector<vector<int>> mx;

gp_hash_table<pii, pair<int, vector<int>>, pair_hash> mem_get;

pair<int, vector<int>> get(int j, int mask) {
    if (mem_get.find({j, mask}) != mem_get.end())
        return mem_get[{j, mask}];

    vector<int> dp0(h);
    vector<bool> p0(h);
    dp0[0] = 0;

    for (int i = 1; i < h; ++i) {
        dp0[i] = dp0[i - 1];
        if ((mask & (1 << i)) || (mask & (1 << (i - 1)))) continue;
        dp0[i] = max(dp0[i], (i - 2 < 0 ? 0 : dp0[i - 2]) + mx[i - 1][j]
   * mx[i][j]);

        if (dp0[i] != dp0[i - 1])
           p0[i] = 1;
    }

    if ((mask & 1) || (mask & (1 << (h - 1)))) {
        vector<int> res;
        int cur = h - 1;
        while (cur > 0) {
            if (p0[cur]) {
                res.pb(cur);
                --cur;
            }
            --cur;
        }
        return mem_get[{j, mask}] = {dp0[h - 1], res};
    }

    int was_mask = mask;

    mask ^= 1;
    mask ^= 1 << (h - 1);

    vector<int> dp1(h);
```

```
    vector<bool> p1(h);
    dp1[0] = mx[0][j] * mx[h - 1][j];
    for (int i = 1; i < h; ++i) {
        dp1[i] = dp1[i - 1];
        if ((mask & (1 << i)) || (mask & (1 << (i - 1)))) continue;
        dp1[i] = max(dp1[i], (i - 2 < 0 ? 0 : dp1[i - 2]) + mx[i - 1][j]
   * mx[i][j]);
        if (dp1[i] != dp1[i - 1])
            p1[i] = 1;
    }

    if (dp0[h - 1] >= dp1[h - 1]) {
        vector<int> res;
        int cur = h - 1;
        while (cur > 0) {
            if (p0[cur]) {
                res.pb(cur);
                --cur;
            }
            --cur;
        }
        return mem_get[{j, was_mask}] = {dp0[h - 1], res};
    } else {
        vector<int> res;
        int cur = h - 1;
        while (cur > 0) {
            if (p0[cur]) {
                res.pb(cur);
                --cur;
            }
            --cur;
        }
        res.pb(0);
        return mem_get[{j, was_mask}] = {dp1[h - 1], res};
    }
}

int ans = 0;
vector<vector<char>> mx_ans;

const int TL = 1.95;

int cnt0 = 0;
bool get_back() {
    ++cnt0;
    if (cnt0 == 100) {
        if (ld(clock() - start) / CLOCKS_PER_SEC >= TL) {
            return 1;
        }
        cnt0 = 0;
    }
    return 0;
}
```

```
gp_hash_table<pii, int, pair_hash> mem;

vector<int> max_pref;

void ans_upd(int cur, vector<vector<char>> &mx_cur) {
    if (cur > ans) {
        ans = cur;
        mx_ans = mx_cur;
    }
}

int rec(int j, int mask, int cur, vector<vector<char>> &mx_cur) {
    if (j == w) return cur;
    if (mem.find({j, mask}) != mem.end() && cur <= mem[{j, mask}]) return
     mem[{j, mask}];
    if (mem.find({j, 0}) != mem.end() && cur <= mem[{j, 0}]) return mem[{
    j, mask}] = cur;
    for (int k = 0; k < j; ++k)
        if (cur < max_pref[k])
            return mem[{j, mask}] = cur;
    for (int mask2 = mask; mask2 > 0; mask2 = (mask2 - 1) & mask)
        if (mem.find({j, mask2}) != mem.end() && cur <= mem[{j, mask2}])
    return mem[{j, mask}] = cur;

    if (j != w - 1) {
        vector<int> good;
        for (int i = 0; i < h; ++i) {
            if (mask & (1 << i)) continue;
            good.pb(i);
        }

        int nxt_mask, cur_mask;

        for (int mask2 = 0; mask2 < (1 << sz(good)); ++mask2) {
            if (get_back()) {
                ans_upd(cur, mx_cur);
                cout << ans << '\n';
                for (auto &el : mx_ans) {
                    for (auto now : el) cout << now;
                    cout << '\n';
                }
                exit(0);
            }

            nxt_mask = 0, cur_mask = mask;
            for (int i = 0; i < sz(good); ++i)
                if (mask2 & (1 << i)) {
                    mx_cur[i][j] = 'L';
                    mx_cur[i][j + 1] = 'R';
                    nxt_mask += (1 << good[i]);
                    cur_mask += (1 << good[i]);
                }
```

```
            pair<int, vector<int>> lol = get(j, cur_mask);

            cur += lol.ff;

            for (auto i : lol.sc) {
                mx_cur[i][j] = 'B';
                mx_cur[(i + 1) % h][j] = 'T';
            }

            mem[{j, mask}] = max(mem[{j, mask}], rec(j + 1, nxt_mask, cur
    , mx_cur));

            for (auto i : lol.sc) {
                mx_cur[i][j] = '.';
                mx_cur[(i + 1) % h][j] = '.';
            }

            cur -= lol.ff;

            for (int i = 0; i < sz(good); ++i)
                if (mask2 & (1 << i)) {
                    mx_cur[i][j] = '.';
                    mx_cur[i][j + 1] = '.';
                }
        }
    } else {
        pair<int, vector<int>> lol = get(j, mask);
        cur += lol.ff;
        for (auto i : lol.sc) {
            mx_cur[i][j] = 'B';
            mx_cur[(i + 1) % h][j] = 'T';
        }
        mem[{j, mask}] = cur;

        // cout << j << ' ' << mask << ": " << get(j, mask).ff << " | "
    << cur << endl;

        ans_upd(cur, mx_cur);
    }

    max_pref[j] = max(max_pref[j], mem[{j, mask}]);
    return mem[{j, mask}];
}

signed main() {
    fast
#ifdef LOCAL
    file_in
#endif

    start = clock();
```

```
    cin >> h >> w;

    max_pref.resize(w);

    mx.resize(h, vector<int>(w));

    for (int i = 0; i < h; ++i) {
        for (int j = 0; j < w; ++j) {
            char c;
            cin >> c;
            mx[i][j] = c - '0';
        }
    }

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

    rec(0, 0, 0, lol);

    cout << ans << '\n';
    for (auto &el : mx_ans) {
        for (auto now : el) cout << now;
        cout << '\n';
    }

    return 0;
}
```