Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	6	0	406

Task A ()

```
/*
 *   cCreated by Kazalika   for Spbu 19-20
 */

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double db;
typedef long double ldb;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vint;
typedef vector<ll> vll;
#define forn(a, i, n) for (int i = a; i < n; ++i)
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define sz(x) int(x.size())
const int N = 2e5 + 5, LOG = 20, MOD = 1e9 + 7;
const ll INF = 1e15;
const ldb EPS = 1e-8;


struct Fenwick {
    int size;
    vll p;
    Fenwick(int size) : size(size) {
        p.resize(size);
    }
    ll get(int l) {
        ll r = 0;
        for (int i = l; i >= 0; i = (i & (i + 1)) - 1)
            r += p[i];
        return r;
    }
    ll get(int l, int r) {
        return get(r) - get(l - 1);
    }

    void add(int ind, ll a){
        for (int i = ind; i < size; i |= i + 1)
            p[i] += a;
    }
};

int main() {
    ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);

    ll n;
    cin >> n;
    cout << n - 1;

}
```

Task B ()

```
/*
 *   cCreated by Kazalika   for Spbu 19-20
 */

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double db;
typedef long double ldb;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vint;
typedef vector<ll> vll;
#define forn(a, i, n) for (int i = a; i < n; ++i)
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define sz(x) int(x.size())
const int N = 2e5 + 5, LOG = 20, MOD = 1e9 + 7;
const ll INF = 1e15;
const ldb EPS = 1e-4;


struct Fenwick {
    int size;
    vll p;
    Fenwick(int size) : size(size) {
        p.resize(size);
    }
    ll get(int l) {
        ll r = 0;
        for (int i = l; i >= 0; i = (i & (i + 1)) - 1)
            r += p[i];
        return r;
    }
    ll get(int l, int r) {
        return get(r) - get(l - 1);
    }

    void add(int ind, ll a){
        for (int i = ind; i < size; i |= i + 1)
            p[i] += a;
    }
};

ldb sq(ldb a) {
    return a * a;
}

struct point {
    ll x, y;
};

point operator- (point a, point b) {
    a.x -= b.x;
    a.y -= b.y;
    return a;
}
point operator+ (point a, point b) {
    a.x += b.x;
    a.y += b.y;
    return a;
}

ldb getLen(point a) {
    return sqrt(sq(a.x) + sq(a.y));
}

void norm(point & a) {
    ldb l = getLen(a);
    a.x /= l; a.y /= l;
}
```

```
void mult(point& a, ldb len) {
    a.x *= len;
    a.y *= len;
}

ll operator% (point a, point b) {
    return a.x * b.y - a.y * b.x;
}

bool half(point a) {
    return a.y > -EPS || abs(a.y) < EPS && a.x > -EPS;
}



ldb dist(point a, point b) {
    return sqrt(sq(a.x - b.x) + sq(a.y - b.y));
}

int main() {
    ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cout.precision(20);

    int n;
    cin >> n;
    if (n == 6) {
        vector <point> v(n);
        forn(0, i, 6) {
            ldb x, y;
            cin >> x >> y;
            x *= 6000; y *= 6000;
            v[i] = {(ll)x, (ll)y};
        }
        ll sx = 0, sy = 0;
        forn(0, i, n) {
            sx += v[i].x;
            sy += v[i].y;
        }

        point c = {sx / 6, sy / 6 };

        stable_sort(all(v), [&](point a, point b) {
            if (half(a - c) != half(b - c)) return half(a - c) > half(b - c);
            return (a - c) % (b - c) > 0;
        });

        cout << ldb(v[0].x) / 6000 << '␣' << ldb(v[0].y) / 6000 << '\n' << ldb(v[1].x) / 6000 << '
            ␣' << ldb(v[1].y) / 6000 << '\n' << ldb(c.x) / 6000 << '␣' << ldb(c.y) / 6000;

    } else {
        vector <point> v(3);
        forn(0, i, 3) {
            ldb x, y;
            cin >> x >> y;
            x *= 6000; y *= 6000;
            v[i] = {(ll)x, (ll)y};
        }
        vector <point> res = { v[0], v[1] };
        res.push_back(v[2] + v[1] - v[0]);
        res.push_back(v[2] + v[2] - v[0]);
        res.push_back(v[2] + v[2] - v[1]);
        res.push_back(v[2] + v[0] - v[1]);
        forn(0, i, 6) {
            cout << ldb(res[i].x) / 6000 << '␣' << ldb(res[i].y) / 6000 << '\n';
        }
    }


}
```

Task C ()

```
/*
 *   cCreated by Kazalika   for Spbu 19-20
 */

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double db;
typedef long double ldb;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vint;
typedef vector<ll> vll;
#define forn(a, i, n) for (int i = a; i < n; ++i)
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define sz(x) int(x.size())
const int N = 1e4 + 5, M = 511, LOG = 20, MOD = 1e9 + 7;
const ll INF = 1e15;
const ldb EPS = 1e-4;
const ll base = 257;
ll powers[N];

ll mup(ll a) {
    return ((a % MOD) + MOD) % MOD;
}

vector <ll> getHash(string& s) {
    vector <ll> h(sz(s));
    h[0] = s[0] - 'a' + 1;
    forn(1, i, sz(s))
        h[i] = (h[i - 1] * base + s[i] - 'a' + 1) % MOD;
    return h;
}

ll getHS(vector<ll>& h, int l, int r) {
    if (l == 0) return h[r];
    return mup(h[r] - h[l - 1] * powers[r - l + 1]);
}

struct Fenwick {
    int size;
    vll p;
    Fenwick(int size) : size(size) {
        p.resize(size);
    }
    ll get(int l) {
        ll r = 0;
        for (int i = l; i >= 0; i = (i & (i + 1)) - 1)
            r += p[i];
        return r;
    }
    ll get(int l, int r) {
        return get(r) - get(l - 1);
    }

    void add(int ind, ll a){
        for (int i = ind; i < size; i |= i + 1)
            p[i] += a;
    }
};

int dp[N][M][2];

int main() {
    ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cout.precision(20);

    string t;
    cin >> t;
    int n;
    cin >> n;
    int res = 0;
    forn(0, q, n) {
        string s;
        cin >> s;
        forn(0, i, sz(s) + 1) {
            forn(0, j, sz(t) + 1)
                dp[i][j][0] = dp[i][j][1] = 1e9;
        }
        forn(0, i, sz(t) + 1) {
            dp[sz(s)][i][0] = sz(t) - i;
        }
        for (int i = 0; i < sz(s); ++i)
            dp[i][sz(t)][0] = 0;
        for (int i = sz(s) - 1; i >= 0; --i) {
            for (int j = sz(t) - 1; j >= 0; --j) {
                if (s[i] == t[j]) {
                    dp[i][j][1] = min(dp[i + 1][j + 1][0], dp[i + 1][j + 1][1]);
                }
                dp[i][j][0] = min(dp[i][j + 1][0] + 1, dp[i][j + 1][1] + 1);
            }
        }
        int mx = 1e9;
        for (int i = 0; i < sz(s) + 1; ++i) {
            mx = min({mx, dp[i][0][0], dp[i][0][1]});
        }
        res += mx;
    }
    cout << res;

}
```

Task D ()

```
/*
 *   cCreated by Kazalika   for Spbu 19-20
 */

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double db;
typedef long double ldb;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vint;
typedef vector<ll> vll;
#define forn(a, i, n) for (int i = a; i < n; ++i)
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define sz(x) int(x.size())
const int N = 2e3 + 5, LOG = 20, MOD = 1e9 + 7;
const ll INF = 1e15;
const ldb EPS = 1e-4;


struct Fenwick {
    int size;
    vll p;
    Fenwick(int size) : size(size) {
        p.resize(size);
    }
    ll get(int l) {
        ll r = 0;
        for (int i = l; i >= 0; i = (i & (i + 1)) - 1)
            r += p[i];
        return r;
    }
    ll get(int l, int r) {
        return get(r) - get(l - 1);
    }

    void add(int ind, ll a){
        for (int i = ind; i < size; i |= i + 1)
            p[i] += a;
    }
};

int rarr[N][N], carr[N][N];
int n, m;


int dist[N][N], distp[N][N];
int dx[8] = { 1, 1, 0, -1, -1, -1, 0, 1 };
int dy[8] = { 0, -1, -1, -1, 0, 1, 1, 1 };
int sd = 8;

bool chk(int a, int b) {
    return a >= 0 && b >= 0 && a < n && b < m;
}

struct pt {
    int x, y;
    bool pr;
    bool operator< (const pt& a) const {
        if (this->x != a.x) return a.x < this->x;
        if (this->y != a.y) return a.y < this->y;
        return a.pr < this->pr;
    }
};


int main() {
    ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cout.precision(20);
    cin >> n >> m;
    pt A, B;
    cin >> A.x >> A.y >> B.x >> B.y;
    A.x--; A.y--; B.x--; B.y--;
    A.pr = true;
    B.pr = true;
    forn(0, i, n) {
        forn(0, j, m) {
            dist[i][j] = distp[i][j] = 1e9;
            int r, c;
            cin >> r >> c;
            rarr[i][j] = r; carr[i][j] = c;
        }
    }
    set <pair<int, pt>> st;
    dist[B.x][B.y] = distp[B.x][B.y] = 0;
    st.insert({0, B});
    st.insert({0, {B.x, B.y, false}});
    while (!st.empty()) {
        pt v = (*(st.begin())).second;
        st.erase(st.begin());
        if (v.pr) {
            for (int i = 0; i < sd; ++i) {
                int xt = v.x + dx[i];
                int yt = v.y + dy[i];
                if (!chk(xt, yt)) continue;
                if (xt + rarr[xt][yt] == v.x && yt + carr[xt][yt] == v.y && dist[xt][yt] > dist[v.x][v.y]) {
                    dist[xt][yt] = dist[v.x][v.y];
                    st.insert({dist[xt][yt], {xt, yt, true}});
                }
                if (dist[v.x][v.y] + abs(v.x - xt - rarr[xt][yt]) + abs(v.y - yt - carr[xt][yt]) < dist[xt][yt]) {
                    dist[xt][yt] = dist[v.x][v.y] + abs(v.x - xt - rarr[xt][yt]) + abs(v.y - yt - carr[xt][yt]);
                    st.insert({dist[xt][yt], {xt, yt, true}});
                }
                if (dist[v.x][v.y] + abs(v.x - xt) + abs(v.y - yt) < distp[xt][yt]) {
                    distp[xt][yt] = dist[v.x][v.y] + abs(v.x - xt) + abs(v.y - yt);
                    st.insert({distp[v.x][v.y], {xt, yt, false}});
                }
            }
        } else {
            if (dist[v.x][v.y] > distp[v.x][v.y] + abs(rarr[v.x][v.y]) + abs(carr[v.x][v.y])) {
                dist[v.x][v.y] = distp[v.x][v.y] + abs(rarr[v.x][v.y]) + abs(carr[v.x][v.y]);
                st.insert({dist[v.x][v.y], { v.x, v.y, true}});
            }
            for (int i = 0; i < sd; ++i) {
                int xt = v.x + dx[i];
                int yt = v.y + dy[i];
                if (!chk(xt, yt)) continue;
                if (distp[v.x][v.y] + abs(v.x - xt - rarr[xt][yt]) + abs(v.y - yt - carr[xt][yt]) < dist[xt][yt]) {
                    dist[xt][yt] = distp[v.x][v.y] + abs(v.x - xt - rarr[xt][yt]) + abs(v.y - yt - carr[xt][yt]);
                    st.insert({dist[xt][yt], {xt, yt, true}});
                }
                if (distp[xt][yt] > distp[v.x][v.y] + abs(v.x - xt) + abs(v.y - yt)) {
                    distp[xt][yt] = distp[v.x][v.y] + abs(v.x - xt) + abs(v.y - yt);
                    st.insert({distp[xt][yt], {xt, yt, false}});
                }
            }
        }
    }

    cout << dist[A.x][A.y];

}
```

Task E ()

```
/*
 *   cCreated by Kazalika   for Spbu 19-20
 */

#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef double db;
typedef long double ldb;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef vector<int> vint;
typedef vector<ll> vll;
#define forn(a, i, n) for (int i = a; i < n; ++i)
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define sz(x) int(x.size())
const int N = 2e3 + 5, LOG = 20, MOD = 1e9 + 7;
const ll INF = 1e15;
const ldb EPS = 1e-4;


struct Fenwick {
    int size;
    vll p;
    Fenwick(int size) : size(size) {
        p.resize(size);
    }
    ll get(int l) {
        ll r = 0;
        for (int i = l; i >= 0; i = (i & (i + 1)) - 1)
            r += p[i];
        return r;
    }
    ll get(int l, int r) {
        return get(r) - get(l - 1);
    }

    void add(int ind, ll a){
        for (int i = ind; i < size; i |= i + 1)
            p[i] += a;
    }
};




int main() {
    ios_base::sync_with_stdio(false); cin.tie(0); cout.tie(0);
    cout.precision(20);

    int n, m, b;
    cin >> n >> m >> b;
    vector <pii> v(b);
    forn(0, i, b) cin >> v[i].first >> v[i].second;
    if (b == 1) {
        cout << "?␣" << v[0].first << '␣' << v[0].second << '␣' << v[0].first + n << '␣' << v[0].
            second + m << endl;
        int x, y;
        cin >> x >> y;
        if (x == v[0].first && y == v[0].second)
            cout << "!␣" << n + 1 << '␣' << m + 1 << endl;
        else
            cout << "!␣" << 1 << '␣' << 1 << endl;
        return 0;
    }

}
```

Task F ()