Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	100	5	425

Task A ()

```cpp
#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

typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()

signed main() {
    fastio;
    #ifdef FlameDragon
        freopen("in.txt", "r", stdin);
    #endif // FlameDragon

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

}
```

Task B ()

```
#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
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio      ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()


const double EPS = 1e-7;
const double PI = atan2(0, -1);
struct vec {
    double x, y;
    vec(){}
    vec(double x, double y) : x(x), y(y){}
    double len() {
        return sqrt(x * x + y * y);
    }
};

vec operator - (vec a, vec b) { return vec(a.x - b.x, a.y - b.y);}
vec operator + (vec a, vec b) { return vec(a.x + b.x, a.y + b.y);}
vec operator * (vec a, double k) { return vec(a.x * k, a.y * k);}

istream &operator >>(istream &in, vec &pt) {
    in >> pt.x >> pt.y;
    return in;
}

ostream &operator << (ostream &out, vec &pt) {
    out << pt.x << ' ' << pt.y << '\n';
    return out;
}

double operator ^(vec a, vec b) {
    return a.x * b.y - a.y * b.x;
}

bool check (vector<vec> a) {
    bool ok = 1;
    a.pb(a[0]);
    for (int i= 1; i < sz(a); i++) {
        int cc = 0;
        vec ab = a[i] - a[i - 1];
        for (int j = 0; j < sz(a); j++) {
            if (j == i || j == i - 1)
                continue;
            vec ac = a[j] - a[i];
            cc += (ab ^ ac) < 0;
        }
        ok &= cc == 4 || cc == 0;
    }

    a.pb(a[1]);
    int cnt = 0;
    for (int i = 1; i < sz(a) - 1; i++) {
        vec ab = a[i] - a[i - 1];
        vec bc = a[i + 1] - a[i];
        if ((ab ^ bc) >= -EPS)
            cnt++;
    }
```

```
    return cnt == 6 && ok;
}

double dist(vec a, vec b) {
    vec ab = b - a;
    return ab.len();
}

void rlx (vec &a) {
    double zn = a.len();
    a.x /= zn;
    a.y /= zn;
}

vec get (vec a, vec b) {
    double d = dist(a, b) / 2;
    d *= tan(PI / 6);
    vec m((a.x + b.x) / 2, (a.y + b.y) / 2);
    //cout << "M: " << m;
    vec ab = b - a;
    vec norm(-ab.y, ab.x);
    rlx(norm);
    norm = norm * d;
    vec c = norm + m;
    vec ac = c - a;
    if ((ac ^ ab) < 0) {
        norm = norm * (-1);
        c = norm + m;

    }
    return c;
}

signed main() {
    fastio;
    cout << fixed << setprecision(10);
    #ifdef FlameDragon
        freopen("in.txt", "r", stdin);
    #endif // FlameDragon

    //cout << tan(PI / 6) << '\n';
    int n;
    cin >> n;
    vector<vec> a(n);
    for (int i = 0; i < n; i++)
        cin >> a[i];
    if (n == 6) {
        vector<int> p(n);
        iota(all(p), 0ll);
        int cc = 0;
        do {
            cc++;
            vector<vec> tmp(n);
            for (int i = 0; i < n; i++)
                tmp[i] = a[p[i]];
            if (check(tmp))
                break;
        } while(next_permutation(all(p)));
        //cout << cc << '\n';;
        for (int i = 0; i < n; i += 2)
            cout << a[p[i]];
    } else {
        a.pb(a[0]);
        for (int i = 0; i < 3; i++) {
            cout << a[i];
            vec c = get(a[i], a[i + 1]);
            cout << c;
        }
    }
}
```

Task C ()

```cpp
#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
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio      ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()

signed main() {
    fastio;
    #ifdef FlameDragon
        freopen("in.txt", "r", stdin);
    #endif // FlameDragon

    string t;
    cin >> t;
    int m = sz(t);
    int n;
    cin >> n;
    vector<string> a(n);
    for (int i = 0; i < n; i++)
        cin >> a[i];
    int ans = 0;
    for (auto s:a) {
        int k = sz(s);
        int delta = m;
        for (int i = 0; i < k; i++) {
            int uk = 0;
            int j = 0;
            while (uk < m && j < k) {
                while (uk < m) {
                    if (t[uk] == s[i + j])
                        break;
                    uk++;
                }
                uk++;
                if (uk ==m + 1)
                    break;
                j++;
                delta = min(delta, m - j);
            }
        }
        ans += delta;
    }
    cout << ans << '\n';
}
```

Task D ()

```
#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
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio      ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()

const int N = 1111;
int X[N][N], Y[N][N];
int dist[2][N][N];

const int INF = 1e9 + 123;

int get(int i1, int j1, int i2, int j2) {
    return abs(i1 + X[i1][j1] - i2) + abs(j1 + Y[i1][j1] - j2);
}

signed main() {
    fastio;
    #ifdef FlameDragon
        freopen("in.txt", "r", stdin);
    #endif // FlameDragon

    int n, m;
    cin >> n >> m;
    int sx, sy, ex, ey;
    cin >> sx >> sy >> ex >> ey;
    sx--, sy--, ex--, ey--;

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            cin >> X[i][j] >> Y[i][j];
        }
    }
    if (max(n, m) > 50) {
        if (ey < sy)
            swap(ey, sy);
        vector<int> dp(m + 1, INF);
        dp[ey] = 0;
        for (int i = ey - 1; i >= sy; i--) {
            for (int j = i + 1; j <= ey; j++) {
                dp[i] = min(dp[i], get(0, i, 0, j) + dp[j]);
            }
        }
        assert(dp[sy] != INF);
        cout << dp[sy] << '\n';
        return 0;
    }
    int ans = INF;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < m; j++) {
            dist[0][i][j] = get(i, j, ex, ey);
        }
    }
    dist[0][ex][ey] = 0;
    for (int k = 0; k < 100; k++) {
        for (int i1 = 0; i1 < n; i1++) {
            for (int j1 = 0; j1 < m; j1++) {
                dist[1][i1][j1] = dist[0][i1][j1];
                for (int i2 = 0; i2 < n; i2++) {
```

```
                    for (int j2 = 0; j2 < m; j2++) {
                        if (i1 == i2 && j1 == j2)
                            continue;
                        dist[1][i1][j1] = min(dist[1][i1][j1], dist[0][i2][j2] + get(i1, j1, i2,
                            j2));
                    }
                }
            }
        }
        ans = min(ans, dist[1][sx][sy]);

        swap(dist[0], dist[1]);
    }
    assert(ans != INF);
    cout << ans << '\n';
}
```

Task E ()

```
#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
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio      ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()

const int INF = 1e9 + 123;
const int kek = 10007570;

int cnt = 0;
void qu (int x1, int y1, int x2, int y2) {
    cnt++;
    assert(cnt <= 8191);
    cout << "? " << x1 << " " << y1 << " " << x2 << " " << y2 << endl;
}

int get () {
    int x, y;
    cin >> x >> y;
    return x / kek;
}

int x[15], y[15];
signed main() {
    fastio;
//    #ifdef FlameDragon
//        freopen("in.txt", "r", stdin);
//    #endif // FlameDragon

    int n, m, b;
    cin >> n >> m >> b;

    vector<pair<int, int>> a(b);
    for (int i = 0; i < b; i++) {
        cin >> x[i] >> y[i];
        x[i]--;
        y[i]--;
    }
    vector<int> used(1 << (b));
    for (int i = 0; i < b; i++) {
        vector<int> cur;
        int cc = (1ll << (b - i - 1));
        int cntt = 0;
        for (int j = 0; j < sz(used); j++) {
            if(used[j] || cntt >= cc)
                continue;
            cur.pb(j);
            if (sz(cur) == 2) {
                qu(cur[0] * kek + x[i], cur[0] * kek + y[i], cur[1] * kek + x[i], cur[1] * kek + y
                    [i]);
                cur.clear();
                int id = get();
                assert(id >= 0);
                used[id] = 1;
                cntt++;
            }
        }
    }
    for (int i = 0; i < sz(used); i++) {
```

```
        if (!used[i]) {
            cout << "! " << i * kek << " " << i * kek << endl;
            return 0;
        }
    }
}
```

Task F ()

```cpp
#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
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    ordered_set;

#define forn(i, n) for (int i = 0 ; i < n; i++)
#define fs first
#define sc second
#define all(x) x.begin(), x.end()
#define rall(x) x.rbegin(), x.rend()
#define MP make_pair
#define pb push_back
#define fastio      ios_base::sync_with_stdio(false), cin.tie(0), cout.tie(0)
#define sz(x) (int)x.size()

const int MOD = 1e9 + 7;

signed main() {
    fastio;
    #ifdef FlameDragon
        freopen("in.txt", "r", stdin);
    #endif // FlameDragon

    int n = 4;
    int m = (n * n * n - n) / 6;

    cin >> n >> m;
    if (n == 2) {
        cout << "1";
    } else if (n == 3){
        cout << "0 0 0 3";
    } else if (n == 4){
        cout << "0 0 0 0 0 0 0 0 0 4 12";
    } else if (n == 5) {
        vector<int> ans(m + 1);
        ans[16] = 5;
        ans[20] = 60;
        ans[18] = 60;
        for (int i = 1; i <= m; i++)
            cout << ans[i] << ' ';
    } else if (n == 6) {
        vector<int> ans(m + 1);
        ans[m] = 360;
        ans[25] = 6;
        ans[31] = 360;
        ans[29] = 90;
        for (int i = 1; i <= m; i++)
            cout << ans[i] << ' ';
    }
}
```