

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	6	18	364

Task A ()

```
// #define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
using namespace std;

signed main() {
    #ifdef LOCAL
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        cout << fixed << setprecision(3);
        srand(time(0));
    #else
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        cout << fixed << setprecision(15);
        srand(time(0));
    #endif
    int n;
    cin >> n;
    cout << n - 1;

    #ifdef LOCAL
        cerr << "proc_time:_" << 1.0 * clock() / CLOCKS_PER_SEC << "s_";
    #endif
    return 0;
}
```

Task B ()

```
// #define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
using namespace std;

long double hight = 0;

pair<long double, long double> recover(pair<long double, long double> a, pair<long double, long
double> b, pair<long double, long double> s) {
    long double mx = a.first + b.first;
    mx /= 2;
    long double my = a.second + b.second;
    my /= 2;
    long double vx = mx - s.first;
    long double vy = my - s.second;
    long double len = hypot(vx, vy);
    vx /= len;
    vy /= len;
    vx *= hight + len;
    vy *= hight + len;
    return make_pair(s.first + vx, s.second + vy);
}

signed main() {
#ifdef LOCAL
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
    cout << fixed << setprecision(3);
    srand(time(0));
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    cout.tie(0);
    cout << fixed << setprecision(3);
    srand(time(0));
#endif
    const long double EPS = 1e-2;
    int n;
    cin >> n;
    if (n == 6) {
        vector<pair<long double, long double>> a(6);
        for (int i = 0; i < n; ++i) {
            cin >> a[i].first >> a[i].second;
        }
        long double R = 1e9;
        for (int i = 0; i < n; ++i) {
            for (int j = i + 1; j < n; ++j) {
                R = min(R, hypot(a[j].first - a[i].first, a[j].second - a[i].
                    second));
            }
        }
        // cout << "R = " << R << '\n';
        vector<pair<long double, long double>> q;
        q.emplace_back(a[0]);
        for (int i = 0; i < 5; ++i) {
            for (int j = 0; j < n; ++j) {
                if (abs(hypot(q.back().first - a[j].first, q.back().second - a[j].
                    second) - R) < EPS && find(q.begin(), q.end(), a[j]) == q.end
                    ()) {
                    q.emplace_back(a[j]);
                    break;
                }
            }
        }
        cout << q[0].first << ' ' << q[0].second << '\n' << q[2].first << ' ' << q[2].
            second << '\n' << q[4].first << ' ' << q[4].second << '\n';
    } else {
        vector<pair<long double, long double>> ans(6);
        cin >> ans[0].first >> ans[0].second;
        cin >> ans[2].first >> ans[2].second;
        cin >> ans[4].first >> ans[4].second;
        long double a = hypot(ans[0].first - ans[2].first, ans[0].second - ans[2].second);
```

```

    long double R = a / sqrt(3);
    hight = sqrt(R*R - a*a/4);
    ans[3] = recover(ans[2], ans[4], ans[0]);
    ans[1] = recover(ans[0], ans[2], ans[4]);
    ans[5] = recover(ans[4], ans[0], ans[2]);
    for (auto now : ans) {
        cout << now.first << ' ' << now.second << '\n';
    }
}
#ifdef LOCAL
    cerr << "proc_time: " << 1.0 * clock() / CLOCKS_PER_SEC << "s\n";
#endif
return 0;
}

```

Task C ()

```
// #define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
using namespace std;

#define fi first
#define se second

int fin[501][30];

signed main() {
    #ifdef LOCAL
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        cout << fixed << setprecision(3);
        srand(time(0));
    #else
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        cout << fixed << setprecision(15);
        srand(time(0));
    #endif
    string t;
    cin >> t;
    int T, whole_ans = 0, tl = t.size();
    for (int i = 0; i <= tl; ++i) {
        for (int j = 'a'; j <= 'z'; ++j) {
            fin[i][j - 'a'] = 1e9;
        }
    }
    for (int i = 0; i < tl; ++i) {
        for (int j = 0; j <= i; ++j) {
            fin[j][t[i] - 'a'] = min(fin[j][t[i] - 'a'], i);
        }
    }
    cin >> T;
    while (T --> 0) {
        string s;
        cin >> s;
        int n = s.size();
        bool found = false;
        for (int len = min(n, tl); !found && len > 0; --len) {
            for (int i = 0; !found && i <= n - len; ++i) {
                string check = s.substr(i, len);
                int ptr = -1;
                for (int j = i; j < i + len; ++j) {
                    if (fin[ptr + 1][s[j] - 'a'] == 1e9) break;
                    else ptr = fin[ptr + 1][s[j] - 'a'];
                    if (j == i + len - 1) {
                        found = true;
                        whole_ans += tl - len;
                        // cout << s << " : " << check << " : " << tl - len << '\n';
                        break;
                    }
                }
            }
        }
        if (!found && len == 1) whole_ans += tl;
    }
    cout << whole_ans;
    #ifdef LOCAL
        cerr << "proc_time:_" << 1.0 * clock() / CLOCKS_PER_SEC << "s_";
    #endif
    return 0;
}
```

Task D ()

```
// #define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
using namespace std;

#define int long long
#define fi first
#define se second

signed main() {
    #ifdef LOCAL
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        cout << fixed << setprecision(3);
        srand(time(0));
    #else
        ios_base::sync_with_stdio(false);
        cin.tie(0);
        cout.tie(0);
        cout << fixed << setprecision(15);
        srand(time(0));
    #endif
    int n, m;
    cin >> n >> m;
    pair<int, int> s;
    pair<int, int> f;
    cin >> s.fi >> s.se >> f.fi >> f.se;
    —s.fi; —s.se; —f.fi; —f.se;
    vector<vector<pair<int, int>>> a(n, vector<pair<int, int>> (m));
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            cin >> a[i][j].fi >> a[i][j].se;
        }
    }
    vector<vector<int>> dist(n, vector<int>(m, 1e9));
    vector<vector<bool>> used(n, vector<bool> (m));
    dist[s.fi][s.se] = 0;
    while (true) {
        bool changed = false;
        pair<int, int> cur = {-1, -1};
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                if ((cur.fi == -1 && cur.se == -1) || dist[i][j] < dist[cur.fi][cur.se]) && !used[i][j]) cur = {i, j};
            }
        }
        if (cur.fi == -1 && cur.se == -1) break;
        used[cur.fi][cur.se] = true;
        // cout << "updating from " << cur.fi + 1 << ' ' << cur.se + 1 << '\n';
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                int new_dist = dist[cur.fi][cur.se] + abs(j - cur.se - a[cur.fi][cur.se].se) + abs(i - cur.fi - a[cur.fi][cur.se].fi);
                if (new_dist < dist[i][j]) {
                    // cout << "updated " << i + 1 << ' ' << j + 1 << " with "
                    << new_dist << '\n';
                    dist[i][j] = new_dist;
                    changed = true;
                }
            }
        }
        // if (changed == false) break;
    }
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            cout << "For " << i + 1 << ' ' << j + 1 << " : ";
            cout << dist[i][j] << '\n';
        }
        cout << '\n';
    }
    cout << dist[f.fi][f.se];
    #ifdef LOCAL
        cerr << "proc_time:_" << 1.0 * clock() / CLOCKS_PER_SEC << "s_";
    #endif
}
```

```
    #endif  
    return 0;  
}
```

Task E ()

```
// #define _GLIBCXX_DEBUG
#include <bits/stdc++.h>
using namespace std;

#define fi first
#define se second

signed main() {
    #ifdef LOCAL
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        cout << fixed << setprecision(3);
        srand(time(0));
    #else
        // ios_base::sync_with_stdio(false);
        // cin.tie(0);
        // cout.tie(0);
        cout << fixed << setprecision(15);
        srand(time(0));
    #endif
    int n, m, b;
    cin >> n >> m >> b;
    if (b == 1) {

        int i1, j1;
        cin >> i1 >> j1;
        pair<int, int> a = {i1, j1}, b = {i1, j1 + m};
        vector<vector<int>> q(n + 1, vector<int>(2 * m + 2));
        q[i1][j1] = 1;
        q[i1][j1 + m] = 1;
        cout << "?_ " << a.fi << '_ ' << a.se << '_ ' << b.fi << '_ ' << b.se << endl;
        cin >> i1 >> j1;
        q[i1][j1] = -1;
        bool good = true;
        for (int i = 1; i <= n; ++i) {
            for (int j = 1; j <= m; ++j) {
                if (i == a.fi && j == a.se && q[i][j] != 1) {good = false;}
                if (q[i][j] == -1) good = false;
            }
        }
        if (good) cout << "!_1_1" << endl;
        else cout << "!_ " << 1 << '_ ' << m + 1 << endl;
        return 0;
    }
    vector<vector<bool>> a(n, vector<bool>(m));
    for (int i = 0; i < b; ++i) {
        int x, y;
        cin >> x >> y;
        --x; --y;
        a[x][y] = 1;
    }
    int wr = 1;
    int wl = -1;
    vector<vector<int>> qr(n + 1, vector<int>(1000000));
    vector<vector<int>> ql(n + 1, vector<int>(1000000));
    while (true) {

    }
    #ifdef LOCAL
        cerr << "proc_time:_ " << 1.0 * clock() / CLOCKS_PER_SEC << "s_";
    #endif
    return 0;
}
```

Task F ()

[illegible]


```

        for (int j = i + 1; j < n; ++j) {
            a[i][j] = bits[ptr++];
            a[j][i] = a[i][j];
        }
    }
    dfs(0);
    bool tree = true;
    for (int i = 0; i < n; ++i) {
        tree &= used[i];
    }
    if (!tree) continue;
    int sum_dist = 0;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            used[j] = false;
            depth[j] = 0;
        }
        dfs(i);
        for (int j = i + 1; j < n; ++j) {
            sum_dist += depth[j];
        }
    }
    ans[sum_dist - 1] = (ans[sum_dist - 1] + 1) % MOD;
}
for (auto now : ans) cout << now << '\n';
#ifdef LOCAL
    cerr << "proc_time:\n" << 1.0 * clock() / CLOCKS_PER_SEC << "s\n\n";
#endif
return 0;
}

```