

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	100	0	440

Task A ()

```
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

#define LOCAL
#define int long long

using namespace std;

void setup() {
#ifdef LOCAL
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#else
    ios::sync_with_stdio(0);
    cout.tie(0);
    cin.tie(0);
#endif
}

int min(int a, int b) {
    if (a > b)
        return b;
    return a;
}

signed main() {
    //setup();

    int n;
    cin >> n;

    int res = 0;
    while(true) {
        int k = (n - res) / 2;
        if (min(k + res, n - k) > res)
            res = min(k + res, n - k);
        else if (min(k + res + 1, n - k - 1) > res)
            res = min(k + res + 1, n - k - 1);
        else if (min(k + res - 1, n - k + 1) > res)
            res = min(k + res - 1, n - k + 1);
        else
            break;
    }
    cout << res;
}
```

Task B ()

```
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <algorithm>
#include <math.h>

// #define int long long

#define double long double

using namespace std;

int polu_pl(pair<double, double> a) {
    if (a.second > 0)
        return 2;
    if (a.second == 0 && a.first > 0)
        return 2;
    return 1;
}

double vect(pair<double, double> a, pair<double, double> b) {
    return a.first * b.second - a.second * b.first;
}

const double KKK = sqrt(3) / 2;

signed main() {
    ios::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);

    int n;
    cin >> n;
    if (n == 6) {
        pair<double, double> O = { 0, 0 };
        vector<pair<double, double>> a(n);
        for (int i = 0; i < n; ++i) {
            cin >> a[i].first >> a[i].second;
            O.first += a[i].first / 6, O.second += a[i].second / 6;
        }
        // cout << O.first << ' ' << O.second << '\n';
        sort(a.begin(), a.end(), [O](const pair<double, double> A, const pair<double, double> B) {
            pair<double, double> a = { A.first - O.first, A.second - O.second }, b = { B.first - O.first, B.second - O.second };
            if (polu_pl(a) != polu_pl(b))
                return polu_pl(a) < polu_pl(b);
            return vect(a, b) > 0;
        });
        for (int i = 0; i < n; i += 2) {
            cout << a[i].first << ' ' << a[i].second << '\n';
        }
    }
    else {
        vector<pair<double, double>> z(n);
        pair<double, double> O = { 0, 0 };
        for (int i = 0; i < 3; ++i) {
            /*double a, b;
            cin >> a >> b;
            cout << a << ' ' << b << '\n';
            cout << (a * 0.5 + b * KKK) << ' ' << (a * KKK + b * 0.5) << '\n';*/
            cin >> z[i].first >> z[i].second;
            O.first += z[i].first / 3, O.second += z[i].second / 3;
        }
        // cout << O.first << ' ' << O.second << '\n';
        for (int i = 0; i < n; ++i) {
            double a = z[i].first - O.first, b = z[i].second - O.second;
            cout << z[i].first << ' ' << z[i].second << '\n';
            cout << (a * 0.5 - b * KKK) + O.first << ' ' << (a * KKK + b * 0.5) + O.second << '\n';
        }
    }
}
```

```

    }
}

/*
6
6.000 9.660
11.000 1.000
-4.000 9.660
-9.000 1.000
6.000 -7.660
-4.000 -7.660
*/

```

Task C ()

```
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

#define LOCAL
// #define int long long

using namespace std;

void setup() {
    #ifdef LOCAL
        // freopen("input.txt", "r", stdin);
        // freopen("output.txt", "w", stdout);
    #else
        ios::sync_with_stdio(0);
        cout.tie(0);
        cin.tie(0);
    #endif
}

int min(int a, int b) {
    if (a > b)
        return b;
    return a;
}

int inf = 2e9;

signed main() {
    // setup();

    string t;
    cin >> t;

    int q;
    cin >> q;
    int ans = 0;
    while (q--) {
        string s;
        cin >> s;
        int res = inf;
        for (int i = 0; i <= s.size(); ++i) {
            int l_res = 0;
            int l = 0;
            int j = i - 1;
            if (i != 0 && s[i - 1] == t[0]) {
                ++l;
                ++j;
            }
            while (j < s.size() && l < t.size()) {
                if (s[j] == t[l])
                    ++l, ++j;
                else
                    ++l, ++l_res;
            }
            l_res += t.size() - l;
            res = min(res, l_res);
        }
        ans += res;
    }
    cout << ans;
}
```

Task D ()

```
#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>
#include <ctime>

//define int long long

#pragma optimize("O3");

using namespace std;

int min(int a, int b) {
    if (a > b)
        return b;
    return a;
}

const int inf = 2e9;
const int N = 1e3 + 228;

pair<int, int> a[N][N];
int d[N][N];
int used[N][N];

int dist(pair<int, int> s, pair<int, int> f) {
    pair<int, int> to = { s.first + a[s.first][s.second].first, s.second + a[s.first][s.second].second };
    return abs(f.first - to.first) + abs(f.second - to.second);
}

signed main() {
    for (int i = 0; i < N; ++i) {
        fill(d[i], d[i] + N, inf);
        fill(used[i], used[i] + N, 0);
    }

    int n, m;
    cin >> n >> m;
    pair<int, int> s, f;
    cin >> s.first >> s.second >> f.first >> f.second;
    --s.first, --s.second, --f.first, --f.second;
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j)
            cin >> a[i][j].first >> a[i][j].second;
    }

    d[f.first][f.second] = 0;

    while (true) {
        if (1.0 * clock() / CLOCKS_PER_SEC >= 1.8)
            break;

        pair<pair<int, int>, int> mn = { {0, 0}, inf };
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < m; ++j) {
                if (d[i][j] < mn.second && !used[i][j]) {
                    mn.first = { i, j };
                    mn.second = d[i][j];
                }
            }
        }

        if (mn.second >= inf)
            break;

        pair<int, int> now = mn.first;
        int price = mn.second;
        used[now.first][now.second] = 1;
    }
}
```

```

    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            pair<int, int> _ = { i, j };
            if (dist(_, now) + price < d[i][j])
                d[i][j] = dist(_, now) + price;
        }
    }
}
cout << d[s.first][s.second];
}

```

Task E ()

```

#include <iostream>
#include <vector>
#include <set>
#include <unordered_set>
#include <map>
#include <unordered_map>

#define int long long

using namespace std;

int min(int a, int b) {
    if (a > b)
        return b;
    return a;
}

const int inf = 2e9;
const int N = 5e5;
const int K = 13;

signed main() {
    int n, m;
    cin >> n >> m;
    int k;
    cin >> k;
    vector<pair<int, int>> a(k);
    for (int i = 0; i < k; ++i)
        cin >> a[i].first >> a[i].second;
    vector<pair<int, pair<int, bool>>> pol((1 << K));
    int _ = 0;
    for (auto& i : pol) {
        i.first = _++;
        i.second = { 0, false };
    }

    //cout << pol.size() << '\n';

    int kol = 0;
    pair<int, int> z[2];
    for (int j = 0; j < (k + 1); ++j) {
        for (int i = 0; i < pol.size(); ++i) {
            if (pol[i].second.first == k && !pol[i].second.second) {
                if (kol == 1) {
                    cout << '?' << '\u' << z[0].first << '\u' << z[0].second << '\n';
                    cout << flush;
                    int a, b;
                    cin >> a >> b;
                    pol[a / N].second.second = true;
                }
                if (!pol[i].second.second) {
                    cout << '!' << '\u' << N * pol[i].first + 1 << '\u' << pol[i].first * N + 1 << '
                    \n';
                    cout << flush;
                    return 0;
                }
            }
        }
        if (!pol[i].second.second) {
            z[kol++] = { N * pol[i].first + a[pol[i].second.first].first, N * pol[i].first + a
            [pol[i].second.first].second };
            ++pol[i].second.first;
            if (kol == 2) {
                kol = 0;
                cout << '?' << '\u' << z[0].first << '\u' << z[0].second << '\u' << z[1].first <<
                '\u' << z[1].second << '\n';
                cout << flush;
                int o, p;
                cin >> o >> p;
                //cout << o / N << ' ' << pol[o / N].first << ':' << pol[o / N].second << '\n
                ';
                for (auto& i : pol) {
                    if (o / N == i.first)

```

```

        i.second.second = true;
    }
}
}
vector<pair<int, pair<int, bool>>> kek;
for (int i = 0; i < pol.size(); ++i) {
    if (!pol[i].second.second)
        kek.push_back(pol[i]);
    if (kek.size() >= (1 << (K - j - 1)))
        break;
}
pol = kek;
}
}

```


Task F ()