

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	6	0	346

Task A ()

```
#include <iostream>
#include <stdio.h>
#include <algorithm>
#include <math.h>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <bitset>
#include <ctime>
#include <random>

using namespace std;

#define inf 1'000'000'007
#define eps 1e-9

typedef long long ll;
typedef unsigned u;
typedef long double ld;

int main() {
    ios_base::sync_with_stdio(0);
    cin.sync_with_stdio(0);
    cout.sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cout.precision(9);
    srand(time(0));

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

Task B ()

```
#include <iostream>
#include <stdio.h>
#include <algorithm>
#include <math.h>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <bitset>
#include <ctime>
#include <random>
#include <cassert>

using namespace std;

#define inf 1'000'000'007
#define eps 1e-9

typedef long long ll;
typedef unsigned u;
typedef long double ld;

int main() {
    ios_base::sync_with_stdio(0);
    cin.sync_with_stdio(0);
    cout.sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cout.precision(9);
    srand(time(0));
    int n;
    cin >> n;
    if (n == 6) {
        pair<ld, ld> arr[6];
        for (int i = 0; i < n; i++) {
            cin >> arr[i].first >> arr[i].second;
        }
        vector<int> order;
        bool used[6] = { 0 };
        order.push_back(0);
        used[0] = 1;
        for (int i = 0; i < n - 1; i++) {
            ld d = 1.0 * inf * inf;
            int id = 0;
            ld x = arr[order.back()].first, y = arr[order.back()].second;
            for (int j = 0; j < n; j++) {
                if (!used[j]) {
                    ld nowd = sqrt((x - arr[j].first) * (x - arr[j].first) + (
                        y - arr[j].second) * (y - arr[j].second));
                    if (d > nowd) {
                        d = nowd;
                        id = j;
                    }
                }
            }
            used[id] = 1;
            order.push_back(id);
        }
        for (int i = 0; i < 3; i++) {
            cout << fixed << arr[order[2 * i]].first << "\u" << arr[order[2 * i]].
                second << "\n";
        }
    }
    else {
        pair<ld, ld> arr[3];
        for (int i = 0; i < n; i++) {
            cin >> arr[i].first >> arr[i].second;
        }
    }
}
```

```

    }
    ld x = (arr[0].first + arr[1].first + arr[2].first) / 3.0, y = (arr[0].second +
        arr[1].second + arr[2].second) / 3.0;
    for (int i = 0; i < 3; i++) {
        cout << fixed << arr[i].first << "┘" << arr[i].second << "\n";
        if (i == 0) {
            cout << fixed << x + (x - arr[2].first) << "┘" << y + (y - arr[2].
                second) << "\n";
        }
        else if (i == 1) {
            cout << fixed << x + (x - arr[0].first) << "┘" << y + (y - arr[0].
                second) << "\n";
        }
        else {
            cout << fixed << x + (x - arr[1].first) << "┘" << y + (y - arr[1].
                second) << "\n";
        }
    }
}

return 0;
}

```

Task C ()

```
#include <iostream>
#include <stdio.h>
#include <algorithm>
#include <math.h>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <bitset>
#include <ctime>
#include <random>

using namespace std;

#define inf 1'000'000'007
#define eps 1e-9

typedef long long ll;
typedef unsigned u;
typedef long double ld;

int nxt[27][512];
string s, t;

int main() {
    ios_base::sync_with_stdio(0);
    cin.sync_with_stdio(0);
    cout.sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cout.precision(9);
    srand(time(0));

    cin >> s;
    s = '#' + s;
    for (int i = 0; i < 26; i++) {
        nxt[i][s.size() - 1] = -1;
    }
    for (int i = s.size() - 1; i > 0; i--) {
        for (int j = 0; j < 26; j++) {
            nxt[j][i - 1] = nxt[j][i];
        }
        nxt[s[i] - 'a'][i - 1] = i;
    }

    int q;
    int ans = 0;
    cin >> q;
    while (q--) {
        cin >> t;
        int now = inf;
        for (int i = 0; i < t.size(); i++) {
            int cur = 0;
            int poss = 0, post = i;
            while (poss < s.size() && post < t.size() && nxt[t[post] - 'a'][poss] !=
                -1) {
                poss = nxt[t[post] - 'a'][poss];
                cur++;
                post++;
            }
            now = min(now, (int)s.size() - 1 - cur);
        }
        //cout << now << "\n";
        ans += now;
    }

    cout << ans;
}
```

Task D ()

```
#include <iostream>
#include <stdio.h>
#include <algorithm>
#include <math.h>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <bitset>
#include <ctime>
#include <random>

using namespace std;

#define inf 1'000'000'007
#define eps 1e-9

typedef long long ll;
typedef unsigned u;
typedef long double ld;

ll n, m;
ll sx, sy, fx, fy;
pair<int, int> arr[1024][1024];
int dp[1024][1024];

int main() {
    ios_base::sync_with_stdio(0);
    cin.sync_with_stdio(0);
    cout.sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cout.precision(9);
    srand(time(0));
    cin >> n >> m >> sx >> sy >> fx >> fy;
    --sx, --sy, --fx, --fy;
    for (ll i = 0; i < n; i++) {
        for (ll j = 0; j < m; j++) {
            cin >> arr[i][j].first >> arr[i][j].second;
            dp[i][j] = inf;
        }
    }

    set< pair<int, pair<int, int>>> q;
    dp[sx][sy] = 0;
    q.insert({ 0, {sx, sy} });
    while (q.size()) {
        int x = q.begin()->second.first;
        int y = q.begin()->second.second;
        q.erase(q.begin());
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m; j++) {
                int nowx = x + arr[x][y].first, nowy = y + arr[x][y].second;
                int now = abs(i - nowx) + abs(j - nowy);
                if (dp[x][y] + now < dp[i][j]) {
                    q.erase({ dp[i][j], {i, j} });
                    dp[i][j] = dp[x][y] + now;
                    q.insert({ dp[i][j], {i, j} });
                }
            }
        }
        //continue;
    }
    cout << dp[fx][fy];
}
```

Task E ()

```
#include <iostream>
#include <stdio.h>
#include <algorithm>
#include <math.h>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <unordered_set>
#include <bitset>
#include <ctime>
#include <random>
#include <cassert>

using namespace std;

#define inf 1'000'000'007
#define eps 1e-9

typedef long long ll;
typedef unsigned u;
typedef long double ld;

ll n, m, q;
int num[200005];
pair <ll, ll> arr[16];
int main() {
    ios_base::sync_with_stdio(0);
    cin.sync_with_stdio(0);
    cout.sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
    cout.precision(9);
    srand(time(0));
    cin >> n >> m >> q;
    for (int i = 0; i < q; i++) {
        cin >> arr[i].first >> arr[i].second;
        arr[i].first --, arr[i].second --;
    }
    arr[q] = { -1, -1 };
    set <pair <int, int>> free;
    for (int i = 0; i < 2 * (1 << q); i++) {
        free.insert({ 0, i });
    }
    for (int i = 0; i < 4 * (1 << q); i++) {
        int a = free.begin()->second;
        free.erase(free.begin());
        if (num[a] == q) {
            cout << "!_0_" << a * m << "\n";
            cout.flush();
            return 0;
        }
        int b = free.begin()->second;
        free.erase(free.begin());
        if (num[b] == q) {
            cout << "!_0_" << b * m << "\n";
            cout.flush();
            return 0;
        }
        pair <ll, ll> one, two;
        one = { arr[num[a]].first, arr[num[a]].second + (1LL * a * m) };
        two = { arr[num[b]].first, arr[num[b]].second + (1LL * b * m) };

        cout << "?_" << one.first << "_" << one.second << "_" << two.first << "_" << two.
            second << "\n";
        cout.flush();
        num[a]++, num[b]++;
        int x, y;
```

```

        cin >> x >> y;
        if (num[y / m] != -1) {
            num[y / m] = -1;
        }

        if (num[a] != -1) {
            free.insert({ num[a], a });
        }
        if (num[b] != -1) {
            free.insert({ num[b], b });
        }
    }

    for (int i = 0; i < 2 * (1 << q); i++) {
        if (num[i] == q) {
            cout << "!_0_" << i * m << "\n";
            cout.flush();
            return 0;
        }
    }
    assert(0);
    return 0;
}

```

Task F ()