

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	100	0	400

Task A ()

```
#include <iostream>
#include <algorithm>
#include <utility>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <iomanip>
#include <assert.h>

#define int long long
#define all(a) a.begin(), a.end()
#define pii pair<int, int>

using namespace std;

typedef long long ll;
typedef long double ld;

void speedup() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
}

void solve() {
    int n;
    cin >> n;
    cout << n - 1;
}

signed main() {
    speedup();
    solve();
    return 0;
}
```

Task B ()

```
#include <iostream>
#include <algorithm>
#include <utility>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <iomanip>
#include <assert.h>

#define int long long
#define all(a) a.begin(), a.end()
#define pii pair<int, int>

using namespace std;

typedef long long ll;
typedef long double ld;

void speedup() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
}

struct coords {
    ld x, y;

    coords() {}
    coords(ld _x, ld _y) {
        x = _x, y = _y;
    }
    void input() {
        cin >> x >> y;
    }
    void print() {
        cout << x << " " << y << "\n";
    }
    coords operator - (coords p) {
        return coords(x - p.x, y - p.y);
    }
    coords operator * (ld p) {
        return coords(x * p, y * p);
    }
    coords operator + (coords p) {
        return coords(x + p.x, y + p.y);
    }
};

bool cmp(coords a, coords b) {
    if (a.x == b.x) {
        return a.y < b.y;
    }
    return a.x < b.x;
}

void solve() {
    int k;
    cin >> k;
    cout << fixed << setprecision(20);
    if (k == 6) {
        vector<coords> a(6);
        for (int i = 0; i < 6; ++i) {
            a[i].input();
        }
        sort(all(a), cmp);
    }
}
```

```

        /*for (int i = 0; i < 6; ++i) {
            a[i].print();
        }*/
        a[0].print();
        a[1].print();
        a[5].print();
    }
    if (k == 3) {
        vector <coords> a(6);
        a[0].input();
        a[1].input();
        a[3].input();
        coords s = (a[3] - a[0]) * 0.5;
        a[2] = a[1] + s;
        coords s2 = (a[1] - a[0]) * (-1);
        a[4] = a[3] + s2;
        s = s * (-1);
        a[5] = a[4] + s;
        for (int i = 0; i < 6; ++i) {
            a[i].print();
        }
    }
}

signed main() {
    speedup();
    solve();
    return 0;
}

```

Task C ()

```
#include <iostream>
#include <algorithm>
#include <utility>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <iomanip>
#include <assert.h>
#include <string>

#define int long long
#define all(a) a.begin(), a.end()
#define pii pair<int, int>

using namespace std;

typedef long long ll;
typedef long double ld;

void speedup() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
}

string s;

int findMx(string& f, int x, int y) {
    int k = 0;
    while (x < s.size() && y < f.size()) {
        if (s[x] == f[y]) {
            k++;
            x++;
            y++;
        }
        if (x >= s.size() || y >= f.size())
            break;
        while (x < s.size() && s[x] != f[y]) {
            ++x;
        }
    }
    return k;
}

int get(string& f) {
    int n = s.size();
    int k = 0;
    for (int i = 0; i < n; ++i) {
        vector<int> st;
        for (int j = 0; j < f.size(); ++j) {
            if (f[j] == s[i]) {
                k = max(k, findMx(f, i, j));
            }
        }
    }
    return n - k;
}

void solve() {
    cin >> s;
    int k, ans = 0;
    cin >> k;
    for (int i = 0; i < k; ++i) {
        string f;
        cin >> f;
        ans += get(f);
    }
    cout << ans;
}
```

```
signed main() {  
    speedup();  
    solve();  
    return 0;  
}
```

Task D ()

Task E ()

```
#include <iostream>
#include <algorithm>
#include <utility>
#include <cmath>
#include <vector>
#include <set>
#include <map>
#include <iomanip>
#include <assert.h>
#include <string>

#define int long long
#define all(a) a.begin(), a.end()
#define pii pair<int, int>

using namespace std;

typedef long long ll;
typedef long double ld;

void speedup() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.tie(0);
}

vector<bool> used;
vector<int> kol;
vector<pii> a;

int getind(int x) {
    return x / 300000;
}

int ind = 0;
int n, m, b;
int MAXN;

void get() {
    while (!used[ind]) {
        ind++;
        ind %= (1 << b);
    }
    /*pii fs = { good[ind].first + a[i].first, good[ind].second + a[i].second };
    while (ind < 4096 && !used[i]) {
        ind++;
    }
    if (ind >= 4096)
        break;
    pii sec = { good[ind].first + a[i].first, good[ind].second + a[i].second };
    cout << " " << fs.first << " " << fs.second << " " << sec.first << " " << sec.second <<
        '\n';
    pii bad;
    cin >> bad.first >> bad.second;
    used[getind(bad.first)] = false;*/
}

int get2() {
    int k1 = b + 1, in = 0;
    for (int j = 0; j < MAXN; ++j) {
        if (used[j] && kol[j] < k1) {
            k1 = kol[j];
            in = j;
        }
    }
    return in;
}

void solve() {
    cin >> n >> m >> b;
    for (int i = 0; i < b; ++i) {
```

```

        int x, y;
        cin >> x >> y;
        a.push_back({ x - 1, y - 1 });
    }
    used.resize(1 << b, true);
    kol.resize(1 << b, 0);
    MAXN = (1 << b);
    vector<pii> good;
    for (ll i = 0; i < MAXN * 300000; i += 300000) {
        good.push_back({ i, 0 });
    }
    for (ll i = 1; i < MAXN; ++i) {
        get();
        if (kol[ind] == b) {
            cout << "!_" << good[ind].first << "_" << good[ind].second << endl;
            return;
        }
        pii fs = { good[ind].first + a[kol[ind]].first, good[ind].second + a[kol[ind]].second };
        kol[ind]++;
        ++ind;
        ind %= (1 << b);
        get();
        if (kol[ind] == b) {
            cout << "!_" << good[ind].first << "_" << good[ind].second << endl;
            return;
        }
        pii sec = { good[ind].first + a[kol[ind]].first, good[ind].second + a[kol[ind]].second };
        kol[ind]++;
        ++ind;
        ind %= (1 << b);
        cout << "?_" << fs.first << "_" << fs.second << "_" << sec.first << "_" << sec.second << endl;
        pii bad;
        cin >> bad.first >> bad.second;
        int z = getind(bad.first);
        if (!used[z]) {
            z = get2();
        }
        used[z] = false;
    }
    ind = 0;
    get();
    cout << "!_" << good[ind].first << "_" << good[ind].second << endl;
}

signed main() {
    speedup();
    solve();
    return 0;
}

/*
2 2 3
1 2
2 2
2 1

0 1

900000 1

1500000 1

2100000 1

1800000 1

600001 1
*/

```

Task F ()