

# 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 <algorithm>
#include <iostream>
#include <string>
#include <cmath>
#include <set>
#include <map>

typedef long long ll;

using namespace std;

const ll cst = 1e5 + 5;
const ll inf = (ll)1e18 + 228;
const ll mod = 1e9 + 7;

int main() {
    ll n;
    cin >> n;
    cout << n - 1;
    return 0;
}
```

## Task B ()

```
#include <algorithm>
#include <iostream>
#include <vector>
#include <cmath>
#include <set>
#include <map>

#define all(x) x.begin(), x.end()

typedef long long ll;
typedef double ld;

using namespace std;

const ll cst = 1e5 + 5;
const ll inf = (ll)1e18 + 228;
const ll mod = 1e9 + 7;
const ld eps = 1e-4;

struct pt{
    ld x, y;
    pt(ld nx = 0.0, ld ny = 0.0){
        x = nx;
        y = ny;
    }
    ld operator*(pt that){
        return this->x * that.x + this->y*that.y;
    }
    ld operator^(pt that){
        return this->x * that.y - this->y*that.x;
    }
    pt operator*(ld k){
        return pt(x * k, y * k);
    }
    pt operator/(ld k){
        return pt(x / k, y / k);
    }
    pt operator+(pt that){
        return pt(x + that.x, y + that.y);
    }
    pt operator-(pt that){
        return pt(x - that.x, y - that.y);
    }
    bool operator==(pt that){
        if(abs(x - that.x) < eps && abs(y - that.y) < eps) return true;
        else return false;
    }
    ld len(){
        return sqrt(x * x + y * y);
    }
    ld lenTo(pt that){
        return (*this - that).len();
    }
};

bool dbeq(ld a, ld b){
    return abs(a - b) < eps;
}

bool dble(ld a, ld b){
    return a + eps < b;
}

bool dbbi(ld a, ld b){
    return b + eps < a;
}

vector<pt> vc;

pt ml(inf, inf);

bool cmp(pt a, pt b){
```

```

    if(dbeq(a.x, ml.x)) return 1;
    if(dbeq(b.x, ml.x)) return 0;
    return dbbi((a.y - ml.y) / (a.x - ml.x), (b.y - ml.y) / (b.x - ml.x));
}

int main(){
    cin.tie(0)->sync_with_stdio(0);
    ll n;
    cin >> n;
    for(int i = 0; i < n; i++){
        ld a, b;
        cin >> a >> b;
        vc.push_back({a, b});
    }
    if(n == 6){
        ll mli = 0;
        for(int i = 0; i < 6; i++){
            if(dble(vc[i].x, ml.x)){
                ml = vc[i];
                mli = i;
            }
            else if(dbeq(vc[i].x, ml.x) && dble(vc[i].y, ml.y)) {
                ml = vc[i];
                mli = i;
            }
        }
        swap(vc[0], vc[mli]);
        sort(all(vc), cmp);
        for(int i = 0; i < 6; i += 2){
            cout.precision(4);
            cout << fixed << vc[i].x << " " << vc[i].y << endl;
        }
        return 0;
    }
    for(int i = 0; i < 3; i++){
        pt a = vc[i];
        pt b = vc[(i + 1) % 3];
        pt c = vc[(i + 2) % 3];
        pt mid = (a + b) / 2.0;

        pt vec = mid - c;
        vec = vec * (4.0 / 3.0);
        vec = vec + c;
        vc.push_back(vec);
    }
    ll mli = 0;
    for(int i = 0; i < 6; i++){
        if(dble(vc[i].x, ml.x)){
            ml = vc[i];
            mli = i;
        }
        else if(dbeq(vc[i].x, ml.x) && dble(vc[i].y, ml.y)) {
            ml = vc[i];
            mli = i;
        }
    }
    swap(vc[0], vc[mli]);
    sort(all(vc), cmp);
    for(int i = 0; i < 6; i += 1){
        cout.precision(4);
        cout << fixed << vc[i].x << " " << vc[i].y << endl;
    }
    return 0;
}

/*
3
-10.0000 -0.0000
5.0000 8.6600
5.0000 -8.6600

```



## Task C ()

```
#include <algorithm>
#include <iostream>
#include <string>
#include <cmath>
#include <set>
#include <map>

typedef long long ll;

using namespace std;

const ll cst = 1e5 + 5;
const ll inf = (ll)1e18 + 228;
const ll mod = 1e9 + 7;

string t;

ll ans;

int main(){
    cin.tie(0)->sync_with_stdio(0);
    ll n;
    cin >> t;
    cin >> n;
    for(int _ = 0; _ < n; _++){
        string s;
        cin >> s;
        ll pans = t.size();
        ll cans = 0;
        for(int i = 0; i < s.size(); i++){
            ll pos = i;
            cans = 0;
            for(int j = 0; j < t.size(); j++){
                if(s[pos] == t[j]) pos++;
                else cans++;
            }
            pans = min(pans, cans);
        }
        ans += pans;
    }
    cout << ans;
    return 0;
}
```

## Task D ()

```
#include <algorithm>
#include <iostream>
#include <vector>
#include <cmath>
#include <set>
#include <map>

#define int long long
#define all(x) x.begin(), x.end()

#pragma GCC optimize("Ofast,unroll-loops,fast-math,no-stack-protector")

typedef long long ll;
typedef double ld;

using namespace std;

const ll cst = 1e6 + 5;
const ll inf = (ll)1e18 + 228;
const ll mod = 1e9 + 7;
const ld eps = 1e-4;

struct cr{
    ll x, y, t;
    cr(ll nx = 0, ll ny = 0, ll nt = -1){
        x = nx;
        y = ny;
        t = nt;
    }
    bool operator<(const cr that) const{
        if(x != that.x) return x < that.x;
        if(y != that.y) return y < that.y;
        return t < that.t;
    }
};

ll n, m;

ll sx, sy, fx, fy;

pair<ll, ll> arr[(ll)1e3 + 5][(ll)1e3 + 5];

ll d[(ll)1e3 + 5][(ll)1e3 + 5];

signed main(){
    cin.tie(0)->sync_with_stdio(0);
    cin >> n >> m;
    cin >> sx >> sy >> fx >> fy;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
            cin >> arr[i][j].first >> arr[i][j].second;
        }
    }
    for(int i = 0; i < (ll)1e3 + 5; i++){
        for(int j = 0; j < (ll)1e3 + 5; j++) d[i][j] = inf;
    }
    set<pair<ll, cr>> q;
    d[sx][sy] = 0;
    cr xxx(sx, sy);
    q.insert(make_pair(0ll, xxx));
    while(!q.empty()){
        cr v = (*q.begin()).second;
        q.erase(q.begin());
        for(int dx = -v.x + 1; dx <= n; dx++){
            ll nx = v.x + dx;
            if(nx > n) break;
            for(int dy = -v.y + 1; dy <= m; dy++){
                ll ny = v.y + dy;
                if(ny > m) break;
                if(d[nx][ny] > d[v.x][v.y] + abs(v.x + arr[v.x][v.y].first - nx) + abs(v.y + arr[v.x][v.y].second - ny)){
                    xxx = cr{nx, ny, 0};
                }
            }
        }
    }
}
```

```

        q.erase({d[nx][ny], xxx});
        d[nx][ny] = d[v.x][v.y] + abs(v.x + arr[v.x][v.y].first - nx) + abs(v.y + arr[
            v.x][v.y].second - ny);
        q.insert({d[nx][ny], xxx});
    }
}
ll nx = v.x + arr[v.x][v.y].first;
ll ny = v.y + arr[v.x][v.y].second;
if(nx < 1 || nx > n || ny < 1 || ny > m) continue;
if(d[nx][ny] > d[v.x][v.y]){
    xxx = cr{nx, ny, 0};
    q.erase({d[nx][ny], xxx});
    d[nx][ny] = d[v.x][v.y];
    q.insert({d[nx][ny], xxx});
}
}
cout << d[fx][fy];
return 0;
}

```

## Task E ()

```
#include <algorithm>
#include <iostream>
#include <vector>
#include <cmath>
#include <bitset>
#include <map>

#define int long long
#define all(x) x.begin(), x.end()

#pragma GCC optimize("Ofast,unroll-loops,fast-math,no-stack-protector")

typedef long long ll;
typedef double ld;

using namespace std;

const ll cst = 1e5 + 5;
const ll inf = (1ll)1e18 + 228;
const ll mod = 1e9 + 7;
const ld eps = 1e-4;

bitset<cst> ban;

ll n, m, b;

vector<pair<ll, ll>> pts;

ll getnum(ll x){
    return (x - 1ll) / n;
}

ll cordin(ll x, ll nm){
    return x + nm * n;
}

ll qr = 0;

ll tl(ll x){
    if(x == 0ll || x == 1ll) return 1ll;
    return tl(x - 1ll) + tl(x - 2ll);
}

signed main(){
    cin.tie(0) -> sync_with_stdio(0);
    cin >> n >> m >> b;
    for(int i = 0; i < b; i++){
        ll x, y;
        cin >> x >> y;
        pts.push_back({x, y});
    }
    reverse(all(pts));
    for(int q = b - 1; q >= 0; q--){
        vector<ll> unban;
        ll kek = 0;
        for(int i = 0; i < (1 << b); i++) if(!ban[i]) unban.push_back(i);
        for(int i = 0; i < unban.size() - 1; i += 2){
            qr++;
            if(qr >= (1 << b)) {
                //tl(100);
            }
            cout << "?_" << cordin(pts[q].first, unban[i]) << "_" << pts[q].second << "_" <<
                cordin(pts[q].first, unban[i + 1]) << "_" << pts[q].second << endl;
            ll x, y;
            cin >> x >> y;
            if(ban[getnum(x)] == 1) kek++;
            ban[getnum(x)] = 1;
        }
        for(int i = 0; i < (1 << b); i++) {
            if(!ban[i] && kek > 0){
                kek--;
                ban[i] = 1;
            }
        }
    }
}
```

```

        }
        if(kek <= 0) break;
    }
}
//if(qr != (1 << b) - 1) tl(100);
for(int i = 0; i < (1 << b); i++) {
    if(!ban[i]) {
        cout << "!_ " << 1ll + (1ll)i * n << "_ " << 1ll << endl;
        return 0;
    }
}
return 0;
}

```

## Task F ()