

# Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	6	0	326

## Task A ()

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    ll n;
    cin >> n;
    cout << (2 * n - 1) / 2;
    return 0;
}
```

## Task B ()

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

struct point{
    ld x, y;
};

point su(point a, point b){
    return {a.x + b.x, a.y + b.y};
}

point di(point a, point b){
    return {a.x - b.x, a.y - b.y};
}

bool cmp(point a, point b){
    return a.x == b.x ? a.y < b.y : a.x < b.x;
}

ld dist(point a, point b){
    ld fv = a.x - b.x;
    ld sv = a.y - b.y;
    return fv * fv + sv * sv;
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    int n;
    cin >> n;
    if(n == 6){
        vector<point> kek(6);
        for(int i = 0; i < 6; i++){
            cin >> kek[i].x >> kek[i].y;
        }
        ld best_dif = 1e18;
        int bi = 0, bj = 0, bk = 0;
        cout << fixed << setprecision(10);
        for(int i = 0; i < 6; i++){
            for(int j = 0; j < i; j++){
                for(int k = 0; k < j; k++){
                    ld f = dist(kek[i], kek[j]);
                    ld s = dist(kek[i], kek[k]);
                    ld t = dist(kek[k], kek[j]);
                    ld sm = abs(f - s) + abs(s - t) + abs(t - f);
                    if(sm < best_dif){
                        best_dif = sm;
                        bi = i;
                        bj = j;
                        bk = k;
                    }
                }
            }
        }
        cout << kek[bi].x << " " << kek[bi].y << "\n";
        cout << kek[bj].x << " " << kek[bj].y << "\n";
        cout << kek[bk].x << " " << kek[bk].y << "\n";
        return 0;
        return -1;
    }
    else{
        vector<point> a(3), ans(6);
        for(int i = 0; i < n; i++)cin >> a[i].x >> a[i].y;
        ld len = 2 * hypot(a[0].x - a[1].x, a[0].y - a[1].y) / sqrt(3);
        //cout << len << "\n";
        for(int i = 0; i < 3; i++){
            ans[2 * i] = a[i];
        }
    }
}
```

```

    point cur = su(di(a[(i + 1) % 3], a[i]), di(a[(i + 2) % 3], a[i]));
    cur = su(cur, a[i]);
    //cout << a[i].x << " " << a[i].y << "\n" << cur.x << " " << cur.y << "\n";
    ld B = cur.x - a[i].x;
    ld A = a[i].y - cur.y;
    ld C = cur.y * a[i].x - cur.x * a[i].y;
    //cout << A << " " << B << " " << C << "\n";
    ld nor = sqrt(A * A + B * B);
    A /= nor; B /= nor;
    point res = {a[i].x + B * len, a[i].y - A * len};
    if(i == 0) ans[3] = res;
    else if(i == 1) ans[5] = res;
    else ans[1] = res;
}
for(int i = 0; i < 6; i++){
    cout << fixed << setprecision(8) << ans[i].x << " " << ans[i].y << "\n";
}
}
}

```

## Task C ()

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    string t;
    cin >> t;
    vector<vector<int>> > next(26);
    for(int i = 0; i < t.size(); i++)next[t[i] - 'a'].push_back(i);
    for(int i = 0; i < 26; i++)next[i].push_back(1000000000);
    int n;
    cin >> n;
    vector<string> a(n);
    ll ans = 0;
    for(int i = 0; i < n; i++){
        cin >> a[i];
        int l = 0, r = (int)t.size() + 1;
        while(r - l - 1 > 0){
            int m = (r + l) / 2;
            bool good = false;
            for(int j = 0; j <= (int)a[i].size() - m; j++){
                int cp = -1;
                for(int k = j; k < j + m; k++){
                    int it = *(upper_bound(next[a[i]][k] - 'a'.begin(), next[a[i]][k] - 'a'.end(),
                                            cp));
                    if(it == 1000000000)break;
                    cp = it;
                    if(k == j + m - 1)good = true;
                }
            }
            if(good){
                l = m;
            }
            else r = m;
        }
        ans += (int)t.size() - 1;
    }
    cout << ans;
}
```

## Task D ()

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

int main()
{
    //ios_base::sync_with_stdio(false);
    //cin.tie(0);
    int n, m;
    int xo, yo, xt, yt;
    cin >> n >> m >> xo >> yo >> xt >> yt;
    xo--;yo--;xt--;yt--;
    vector<vector<pair<int, int>>> dir(n, vector<pair<int, int>>> (m));
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            cin >> dir[i][j].first >> dir[i][j].second;
        }
    }
    vector<vector<int>> g(n * m);
    vector<vector<int>> w(n * m, vector<int> (n * m, 1e9));
    vector<int> d(n * m, 1e9);
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            for(int x = 0; x < n; x++){
                for(int y = 0; y < m; y++){
                    if(i == x && j == y){
                        w[i * m + j][x * m + y] = 0;
                        g[i * m + j].push_back(x * m + y);
                        continue;
                    }
                    int x1 = i + dir[i][j].first;
                    int y1 = j + dir[i][j].second;
                    w[i * m + j][x * m + y] = abs(x1 - x) + abs(y1 - y);
                    g[i * m + j].push_back(x * m + y);
                }
            }
        }
    }
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            int x = i + dir[i][j].first, y = j + dir[i][j].second;
            if(x >= 0 && x < n && y >= 0 && y < m && (x != i || j != y)){
                w[i * m + j][x * m + y] = 0;
            }
        }
    }
    d[xo * m + yo] = 0;
    set<pair<int, int>> s;
    s.insert({0, xo * m + yo});
    for(int i = 0; i < n; i++){
        for(int j = 0; j < m; j++){
            if(i == xo && j == yo) continue;
            s.insert({1e9, i * m + j});
        }
    }
    while(!s.empty()){
        pair<int, int> cur = *(s.begin());
        s.erase(s.begin());
        int v = cur.second, we;
        for(int i = 0; i < g[v].size(); i++){
            int u = g[v][i];
            we = w[v][u];
            if(d[v] + we < d[u]){
                s.erase(s.find({d[u], u}));
                //cout << v << " " << u << " " << d[v] << " " << we << " " << d[u] << "\n";
                d[u] = d[v] + we;
                s.insert({d[v] + we, u});
            }
        }
    }
}
```

```
    cout << d[xt * m + yt];  
}
```

## Task E ()

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

int main()
{
    //ios_base::sync_with_stdio(false);
    //cin.tie(0);
    int n, m, b, x1, y1;
    cin >> n >> m >> b >> x1 >> y1;
    cout << "?_ " << 0 << "_ " << 0 << "_ " << 1000001 << "_ " << 1000001 << endl;
    int x, y;
    cin >> x >> y;
    int difx = x1;
    int dify = y1;
    int pos1 = -difx + 1;
    int pos2 = -dify + 1;
    if(x >= pos1 && x <= pos1 + n - 1 && y >= pos2 && y <= pos2 + m - 1){
        cout << "!_ " << 1000001 - difx + 1 << "_ " << 1000001 - dify + 1 << endl;
    }
    else{
        cout << "!_ " << 0 - difx + 1 << "_ " << 0 - dify + 1 << endl;
    }
}
```

## Task F ()