

Олимпиада СПбГУ по информатике 2019/20 учебного года

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

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <math.h>
#include <string.h>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <chrono>
#include <time.h>
#include <random>
using namespace std;

typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef double db;
const int INF = 1000000007;
const ll LLINF = 1000000000000000007LL;
const double EPS = 1e-2;
#ifdef _DEBUG
const int sz = 200;
#else
const int sz = 6000;
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast")
#pragma GCC target("sse,sse2,sse3,ssse3,sse4")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#endif

signed main() {
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

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

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <math.h>
#include <string.h>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <chrono>
#include <time.h>
#include <random>
using namespace std;

typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef double db;
const int INF = 1000000007;
const ll LLINF = 1000000000000000007LL;
const double EPS = 1e-2;
#ifdef _DEBUG
const int sz = 200;
#else
const int sz = 6000;
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast")
#pragma GCC target("sse,sse2,sse3,ssse3,sse4")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#endif

struct point {
    db x, y;
    point() {};
    point(db x, db y) : x(x), y(y) {};

    point operator-(const point &a) const {
        return { x - a.x, y - a.y };
    }

    point operator+(const point &a) const {
        return { x + a.x, y + a.y };
    }

    point operator/(db a) const {
        return { x / a, y / a };
    }

    friend istream& operator>>(istream &in, point &pt) {
        in >> pt.x >> pt.y;
        return in;
    }

    friend ostream& operator<<(ostream &out, point &pt) {
        out << pt.x << ' ' << pt.y << endl;
        return out;
    }
};

db len(point a, point b) {
    a = a - b;
    return sqrt(a.x * a.x + a.y * a.y);
}

signed main() {
```

```

        ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#ifdef _DEBUG
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
#endif

        cout.precision(10);
        int n;
        cin >> n;
        vector<point> p(6);
        for (int i = 0; i < n; ++i) {
            cin >> p[i];
        }
        if (n == 6) {
            vector<int> v = { 0, 1, 2, 3, 4, 5 };
            while (next_permutation(v.begin(), v.end())) {
                point m = (p[v[0]] + p[v[3]]) / 2;
                bool f = 1;
                for (int i = 0; i < n; ++i) {
                    point m1 = (p[v[i]] + p[v[(i + 3) % n]]) / 2;
                    if (abs(m1.x - m.x) > EPS || abs(m1.y - m.y) > EPS) {
                        f = 0;
                        break;
                    }
                    if (abs(len(p[v[i]], p[v[(i + 1) % n]]) - len(p[v[i]], m)) > EPS) {
                        f = 0;
                        break;
                    }
                }
                if (f) break;
            }
            point a = (p[v[0]] + p[v[3]]) / 2;
            /*point m = (p[0] + p[1]) / 2;
            a = (a + m) / 2;*/
            cout << a << p[v[0]] << p[v[1]];
        }
        else {
            /*cout << abs(len(p[0], p[1]) - len(p[2], p[1])) << endl;
            if (abs(len(p[0], p[1]) - len(p[2], p[1])) < EPS) {
                swap(p[0], p[1]);
            }
            else if (abs(len(p[0], p[2]) - len(p[2], p[1])) < EPS) {
                swap(p[0], p[2]);
            }*/
            /*point m = (p[1] + p[2]) / 2;
            p[0] = p[0] + p[0] - m;*/
            p[4] = p[0] + p[0] - p[1];
            p[5] = p[0] + p[0] - p[2];
            point m = (p[2] + p[4]) / 2;
            p[3] = m - p[0] + m;
            p[0] = p[0] + p[0] - m + p[0] - m;
            for (int i = 0; i < 6; ++i) {
                cout << p[i];
            }
        }
    }
}

```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <math.h>
#include <string.h>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <chrono>
#include <time.h>
#include <random>
using namespace std;

typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef double db;
const int INF = 1000000007;
const ll LLINF = 1000000000000000007LL;
const double EPS = 1e-2;
#ifdef _DEBUG
const int sz = 200;
#else
const int sz = 6000;
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast")
#pragma GCC target("sse,sse2,sse3,ssse3,sse4")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#endif

int z[sz];

int z_func(const string& s) {
    int l = 0, r = 0;
    int n = s.size();
    for (int i = 0; i < s.size(); ++i) {
        z[i] = 0;
    }
    z[0] = n;
    int mx = 0;
    for (int i = 1; i < n; ++i) {
        z[i] = (i < r) ? min(z[i - l], r - i) : 0;
        while (i + z[i] < n && s[i + z[i]] == s[z[i]]) {
            z[i]++;
        }
        if (i >= r) {
            l = i;
            r = i + z[i];
        }
        mx = max(mx, z[i]);
    }
    return mx;
}

signed main() {
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    string t;
    cin >> t;
    int n, ans = 0;
```

```

cin >> n;
for (int i1 = 0; i1 < n; ++i1) {
    string s;
    cin >> s;
    vector<int> v(s.size(), -1);
    int mx = 0;
    for (int i = 0; i < s.size(); ++i) {
        int j = 0, k = 0;
        while (i + k < s.size() && j < t.size()) {
            if (s[i + k] == t[j]) {
                k++;
            }
            j++;
        }
        mx = max(mx, k);
    }
    ans += t.size() - mx;
}
cout << ans;
}

```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <math.h>
#include <string.h>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <chrono>
#include <time.h>
#include <random>
using namespace std;

#define int long long

typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef double db;
const int INF = 1000000007;
const ll LLINF = 1000000000000000007LL;
const double EPS = 1e-2;
#ifdef _DEBUG
const int sz = 200;
#else
const int sz = 6000;
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast")
#pragma GCC target("sse,sse2,sse3,ssse3,sse4")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#endif

signed main() {
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    int n, m, si, sj, ei, ej;
    cin >> n >> m >> si >> sj >> ei >> ej;
    si--;
    sj--;
    ei--;
    ej--;
    vector<vector<pii>> v(n);
    vector<vector<int>> d(n);
    for (int i = 0; i < n; ++i) {
        v[i].resize(m);
        d[i].resize(m, INF);
        for (int j = 0; j < m; ++j) {
            cin >> v[i][j].first >> v[i][j].second;
        }
    }
    set<pair<int, pii>> s;
    d[si][sj] = 0;
    s.insert({ 0, {si, sj} });
    while (s.size()) {
        int dist = s.begin()->first;
        int i = s.begin()->second.first;
        int j = s.begin()->second.second;
        s.erase(s.begin());
```

```

        for (int i1 = 0; i1 < n; ++i1) {
            for (int j1 = 0; j1 < m; ++j1) {
                if (d[i1][j1] > dist + abs(i1 - v[i][j].first - i) + abs(j1 - v[i][j].second - j)) {
                    s.erase({ d[i1][j1], {i1, j1} });
                    d[i1][j1] = dist + abs(i1 - v[i][j].first - i) + abs(j1 - v[i][j].second - j);
                    s.insert({ d[i1][j1], {i1, j1} });
                }
            }
        }
        cout << d[ei][ej];
    }

//
//
//
//signed main() {
//    int n, m, si, sj, ei, ej;
//    cin >> n >> m;
//    vector<vector<pii>> v(n);
//    vector<vector<int>> d(n);
//    for (int i = 0; i < n; ++i) {
//        v[i].resize(n);
//        d[i].resize(n, -1);
//        for (int j = 0; j < m; ++j) {
//            cin >> v[i][j].first >> v[i][j].second;
//        }
//    }
//    d[ei][ej] = 0;
//    set<pii> s;
//    s.insert({ ei, ej });
//    vector<int> goi = {-1, -1, -1, 0, 1, 1, 1, 0};
//    vector<int> goj = {-1, 0, 1, 1, 1, 0, -1, -1};
//    while (s.size()) {
//        int i = s.begin()->first;
//        int j = s.begin()->second;
//        s.erase(s.begin());
//
//        for (int k = 0; k < 8; ++k) {
//
//        }
//    }
//}

```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <stdio.h>
#include <stdlib.h>
#include <algorithm>
#include <math.h>
#include <string.h>
#include <string>
#include <vector>
#include <stack>
#include <queue>
#include <set>
#include <map>
#include <unordered_map>
#include <chrono>
#include <time.h>
#include <random>
using namespace std;

#define int long long

typedef long long ll;
typedef unsigned long long ull;
typedef pair<int, int> pii;
typedef pair<ll, ll> pll;
typedef double db;
const int INF = 1000000007;
const ll LLINF = 1000000000000000007LL;
const double EPS = 1e-2;
#ifdef _DEBUG
const int sz = 200;
#else
const int sz = 6000;
#pragma GCC optimize("O3")
#pragma GCC optimize("Ofast")
#pragma GCC target("sse,sse2,sse3,ssse3,sse4")
#pragma GCC optimize("unroll-loops")
#pragma GCC optimize("unswitch-loops")
#endif

signed main() {
    ios_base::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#ifdef _DEBUG
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif

    int n, a, b;
    cin >> a >> b >> n;
    vector<pii> v(n);
    for (int i = 0; i < n; ++i) {
        cin >> v[i].first >> v[i].second;
    }
    set<int> s;
    for (int i = 0; i < (1 << n); ++i) {
        s.insert(i);
    }
    for (int i = 0; i < n; ++i) {
        set<int> ss;
        vector<int> d;
        for (int j = 0; j < (1 << (n - i - 1)); ++j) {
            int k = *s.begin();
            s.erase(s.begin());
            cout << "?_ " << a * k + v[i].first << '_ ' << v[i].second << '_';
            ss.insert(k);
            k = *s.begin();
            ss.insert(k);
            s.erase(s.begin());
            cout << a * k + v[i].first << '_ ' << v[i].second << endl;
            int x, y;
```



```

        cin >> x >> y;
        k = (x - 1) / a;
        d.push_back(k);
    }
    for (int j = 0; j < d.size(); ++j) {
        ss.erase(d[j]);
    }
    s = ss;
}
cout << "!␣" << a * (*s.begin()) + 1 << "␣" << 1 << endl;
}

```

```

/*int n, m, si, sj, ei, ej;
cin >> n >> m;
vector <vector <pii>> v(n);
vector <vector <int>> d(n);
for (int i = 0; i < n; ++i) {
    v[i].resize(n);
    d[i].resize(n, -1);
    for (int j = 0; j < m; ++j) {
        cin >> v[i][j].first >> v[i][j].second;
    }
}
d[ei][ej] = 0;
set <pii> s;
s.insert({ ei, ej });
vector <int> gox()
while (s.size()) {
    int x = s.begin()->first;
    int y = s.begin()->second;
    s.erase(s.begin());
}
}*/

```

Task F ()