

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	24	0	364

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <string.h>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <stdlib.h>
#define int long long
using namespace std;

signed main(){
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    int n;
    cin >> n;
    cout << n-1;
    return 0;
}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <string.h>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <stdlib.h>
#define int long long
#define dob long double
using namespace std;
int n;
dob DIST(pair<int, int>p1, pair<int, int>p2){
    return (p1.first - p2.first)*(p1.first - p2.first) + (p1.second - p2.second)*(p1.second -
        p2.second);
}
signed main(){
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    cout.precision(40);
    cin >> n;
    if (n == 6){
        vector<pair<dob, dob>>v;
        for (int i = 0; i < 6; ++i){
            dob x, y;
            cin >> x >> y;
            v.push_back({ x, y });
        }
        for (int i = 0; i < 5; ++i){
            for (int j = i + 1; j < 6; ++j){
                if (DIST(v[i], v[j])<DIST(v[i], v[i+1])){
                    swap(v[i+1], v[j]);
                }
            }
        }
        cout << v[0].first << '\n' << v[0].second << '\n';
        cout << v[1].first << '\n' << v[1].second << '\n';
        cout << (v[0].first + v[3].first) / 2 << '\n' << (v[0].second + v[3].second) / 2;
        return 0;
    }
    dob x1, y1, x2, y2, x3, y3, x4, y4, x0, y0, x5, y5, dx, dy;
    cin >> x1 >> y1 >> x2 >> y2 >> x0 >> y0;
    x4 = 2 * x0 - x2;
    x3 = 2 * x0 - x1;
    y4 = 2 * y0 - y2;
    y3 = 2 * y0 - y1;
    cout << x1 << '\n' << y1 << '\n';

    cout << x2 << '\n' << y2 << '\n';

    dx = x0 - x3, dy = y0 - y3;
    cout << x2 - dx << '\n' << y2 - dy << '\n';

    cout << x3 << '\n' << y3 << '\n';

    cout << x4 << '\n' << y4 << '\n';

    dx = x0 - x4, dy = y0 - y4;
    cout << x1 - dx << '\n' << y1 - dy << '\n';
}
```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <string.h>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <stdlib.h>
#define int long long
#define dob long double
using namespace std;
const int N = 11111;
string t;
int res,m;
int f(string s){
    int mx = 0;
    for (int i = 0; i < s.size(); ++i){
        int j = i,k=m-1,r=0;
        while (k >= 0 && j >= 0){
            if (t[k] == s[j]){
                r++;
                k--;
                j--;
            }
            else{
                k--;
            }
        }
        mx = max(mx, r);
    }
    return mx;
}
signed main(){
#ifdef DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
    cin >> t; m = t.size();
    int q;
    cin >> q;
    while (q--){
        string s;
        cin >> s;
        res += m - f(s);
    }
    cout << res;
}
```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <string.h>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <stdlib.h>
#include <queue>

#define int long long
#define dob long double
using namespace std;
const int N = 1111;
int n, m, a[N][N][2], sx, sy, fx, fy, w[N][N];
bool used[N][N];
signed main(){
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif

    cin >> n >> m;
    cin >> sx >> sy >> fx >> fy; sx--; sy--; fx--; fy--;
    for (int i = 0; i < n; ++i){
        for (int j = 0; j < m; ++j){
            w[i][j] = 1e9;
            cin >> a[i][j][0] >> a[i][j][1];
        }
    }
    w[sx][sy] = 0;
    priority_queue<pair<int, pair<int, int>>>q;
    q.push({ 0, { sx, sy } });
    while (q.size()){
        int x = q.top().second.first, y = q.top().second.second;
        q.pop();
        if (used[x][y]) continue;
        used[x][y] = 1;
        if (used[fx][fy]) break;
        for (int i = 0; i < n; ++i){
            for (int j = 0; j < m; ++j){
                int d = abs(i - x - a[x][y][0]) + abs(j - y - a[x][y][1]);
                if (w[i][j] > w[x][y] + d){
                    w[i][j] = w[x][y] + d;
                    q.push({ -w[i][j], { i, j } });
                }
            }
        }
    }
    cout << w[fx][fy];
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <algorithm>
#include <vector>
#include <string>
#include <string.h>
#include <map>
#include <unordered_map>
#include <set>
#include <unordered_set>
#include <stdlib.h>
#include <queue>

#define int long long
#define dob long double
using namespace std;
const int N = 4e5+5;
int n, m, B, d[N], used[N];
vector<pair<int, int>>vec;
signed main(){
    cin >> n >> m >> B;
    for (int i = 0; i < B; ++i){
        int x, y;
        cin >> x >> y;
        vec.push_back({ x, y });
    }
    d[0] = 0;
    for (int i = 1; i < N; ++i)d[i] = d[i - 1] + N;
    for (int i = 0; i < B; ++i){
        int x = vec[i].first, y = vec[i].second;
        vector<pair<int, int>> que, bad;
        for (int j = 0; j <= (1ll << B); j++){
            if (used[j]){
                continue;
            }
            que.push_back({ x, y + d[j] });
        }
        for (int j = 0; j < que.size(); j += 2){
            cout << "?_ " << que[j].first << '_ ' << que[j].second << '_ ' << que[j + 1].
                first << '_ ' << que[j + 1].second << endl;
            int x, y;
            cin >> x >> y;
            bad.push_back({ x, y });
        }
        for (int j = 0; j < bad.size(); ++j){
            used[bad[j].second / N] = 1;
        }
    }
    for (int i = 0; i < N; ++i){
        if (!used[i]){
            cout << "!_1_ " << d[i]+1 << endl;
            return 0;
        }
    }
}
```

Task F ()