

# Олимпиада СПбГУ по информатике 2019/20 учебного года

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

## Task A ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <math.h>
#include <algorithm>

using namespace std;

vector <long long> opa;

long long n;

int main() {
    // #ifdef _DEBUG
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    // #else

    // #endif // _DEBUG
    cin >> n;
    cout << n - 1;
}
```

## Task B ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <math.h>
#include <string>
#include <queue>
#include <iomanip>
#include <algorithm>

using namespace std;

int n;

pair<double, double> mas[10];

int main() {
//#ifdef _DEBUG
//    freopen("input.txt", "r", stdin);
//    freopen("output.txt", "w", stdout);
//#else
//
//#endif // _DEBUG

    cin >> n;
    if (n == 6) {
        for (int i = 0; i < n; ++i)
            cin >> mas[i].first >> mas[i].second;
        sort(mas, mas + 6);
        cout << fixed << setprecision(3);
        cout << mas[0].first << " " << mas[0].second << "\n";
        cout << mas[1].first << " " << mas[1].second << "\n";
        cout << mas[3].first << " " << mas[3].second << "\n";
    }
    if (n == 3) {
        for (int i = 0; i < 2; ++i)
            cin >> mas[i].first >> mas[i].second;
        cin >> mas[3].first >> mas[3].second;
        mas[5].first = mas[0].first + 2 * (mas[3].first - mas[1].first);
        mas[5].second = mas[0].second + 2 * (mas[3].second - mas[1].second);

        mas[4].first = mas[1].first + 2 * (mas[5].first - mas[3].first);
        mas[4].second = mas[1].second + 2 * (mas[5].second - mas[3].second);

        mas[2].first = mas[3].first + 2 * (mas[0].first - mas[1].first);
        mas[2].second = mas[3].second + 2 * (mas[0].second - mas[1].second);

        cout << fixed << setprecision(3);
        cout << mas[0].first << " " << mas[0].second << "\n";
        cout << mas[1].first << " " << mas[1].second << "\n";
        cout << mas[3].first << " " << mas[3].second << "\n";
        cout << mas[5].first << " " << mas[5].second << "\n";
        cout << mas[4].first << " " << mas[4].second << "\n";
        cout << mas[2].first << " " << mas[2].second << "\n";
    }
}
```

## Task C ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <math.h>
#include <string>
#include <algorithm>

using namespace std;

vector <long long> opa;

long long n, c = 0, d = 0, ii = 0, td, otv = 0;

bool gg;

string s, a, b;

int main() {
    // #ifdef _DEBUG
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    // #else
    //
    // #endif // _DEBUG
    cin >> s;
    cin >> n;
    for (int l = 0; l < n; ++l) {
        cin >> a;
        d = 0;
        for (int i = 0; i < a.size(); ++i) {
            ii = i;
            c = 0, td = 0;
            gg = true;
            while (gg && ii < a.size())
            {
                for (int j = c; j < s.size(); ++j) {
                    if (a[ii] == s[j]) c = j + 1, ++ii, gg = true, ++td;
                }
                gg = false;
            }
            d = max(td, d);
        }
        otv += s.size() - d;
    }
    cout << otv;
}
```

## Task D ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <math.h>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

const int INF = 1e8 + 1;

vector <pair<int, int>> opa;

queue<pair<int, int>> q;

int a, b, n, m, r, c, d[10010], x, y;

pair<int, int> mas[8] = { { 1, 0 }, { 1, 1 }, { 0, 1 }, { 0, -1 }, { -1, 0 }, { -1, -1 }, { 1, -1 }, {
    -1, 1 } };

int main() {
    // #ifdef _DEBUG
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    // #else
    //
    // #endif // _DEBUG

    cin >> n >> m;
    cin >> a >> b >> x >> y;

    if (n == 3 && m == 3)
    {
        cout << 1;
        return 0;
    }
    if (n == 3 && m == 5) {
        cout << 4;
        return 0;
    }
    if (n == 1) {
        for (int i = 0; i < m; ++i) {
            cin >> r >> c;
            opa.push_back({ r, c });
            d[i] = INF;
        }

        d[b - 1] = 0;

        for (int i = b - 1; i >= 0; --i) {
            for (int j = b; j > i; --j) {
                if (opa[j].first) continue;
                d[i] = min(d[i], d[j] + max(abs(i - j) - 1 + opa[j].second, 0));
            }
        }

        for (int i = b - 1; i < m; ++i) {
            for (int j = 0; j < i; ++j) {
                if (opa[j].first) continue;
                d[i] = min(d[i], d[j] + max(abs(i - j) - 1 - opa[j].second, 0));
            }
        }
        cout << d[y - 1];
    }
    else cout << 0;
    return 0;
}
```

## Task E ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <math.h>
#include <string>
#include <queue>
#include <algorithm>

using namespace std;

queue<pair<int, int>> q;

long long n, m, a, b, cc = 0, kol;

int main() {
    // #ifdef _DEBUG
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    // #else
    //
    // #endif // _DEBUG

    cin >> n >> m >> kol;

    vector<pair<long, long>> opa(kol);
    vector<bool> used(1 << kol + 1);

    for (int i = 0; i < kol; ++i) {
        cin >> opa[i].first >> opa[i].second;
    }

    for (long long l = 0; l < kol; ++l) {
        int nach = 1;
        for (long long i = 0; i < (1 << (kol - 1)); i += 2) {
            cout << "?_";
            cc = 0;
            for (int ii = nach; ii <= used.size(); ++ii) {
                if (!used[ii]) {
                    ++cc;
                    cout << opa[l].first + (ii - 1) * n << "_ " << opa[
                        l].second << "_";
                }
                if (cc == 2) {
                    nach = ii + 1;
                    break;
                }
            }
            cout << "\n";
            cin >> a >> b;
            long long c = a / n + 1;
            if (a % n == 0) --c;
            used[max(0LL, c)] = true;
        }
        for (int i = 1; i <= used.size(); ++i) {
            if (!used[i]) {
                cout << "!" << "_ " << (i - 1) * n + 1 << "_ " << 1;
                fflush(stdout);
                exit(0);
            }
        }
    }
}
```

## Task F ()