

Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	6	18	424

Task A ()

```
#include <iostream>

using namespace std;

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

Task B ()

```
#include <bits/stdc++.h>
#define int long long
// #define _GLIBCXX_DEBUG
typedef long double ld;

using namespace std;

struct pt {
    ld x, y;
    pt(ld x, ld y): x(x), y(y) {}
    pt() {}
};

pt an(pt a) {
    ld sin = 1.0/2;
    ld cos = sqrt(3) / 2;

    sin = -sqrt(3) / 2;
    cos = -1.0 / 2;
    return pt(a.x * cos - a.y * sin, a.x * sin + a.y * cos);
}

pt angle(pt a, pt from) {
    pt nw(a.x - from.x, a.y - from.y);
    nw = an(nw);
    //nw = an(nw);
    //nw = an(nw);
    //nw = an(nw);
    return pt(nw.x + from.x, nw.y + from.y);
}

bool equall(pt a, pt b) {
    const ld EPS = 0.01;
    return fabs(a.x - b.x) <= EPS && fabs(a.y - b.y) <= EPS;
}

ostream& operator << (ostream& out, pt& p) {
    out << setprecision(10) << fixed << p.x << " " << p.y << "\n";
    return out;
}

int32_t main() {
    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
    int n;
    cin >> n;
    if (n == 6) {
        vector<pt> a(n);
        for (auto& i : a) {
            cin >> i.x >> i.y;
        }
        //cout << a[0] << a[1] << a[2];
        //return 0;
        for (int i = 0; i < n; ++i) {
            for (int j = 0; j < n; ++j) {
                for (int k = 0; k < n; ++k) {
                    if (i == j || i == k || j == k) continue;
                    if (equall(angle(a[i], a[j]), a[k])) {
                        cout << a[i] << a[j] << a[k];
                        return 0;
                    }
                }
            }
        }
    }

    pt p = angle(a[3], a[4]);
    cout << setprecision(3) << fixed << p.x << " " << p.y << "\n";
} else {
    vector<pt> a(6);
    for (int i = 0; i < 3; ++i) {
```

```

        cin >> a[i].x >> a[i].y;
    }
    for (int i = 3; i < 6; ++i) {
        a[i] = angle(a[i - 2], a[i - 1]);
    }
    for (int i = 0; i < 6; ++i) {
        cout << a[i];
    }
}
return 0;
}

```

Task C ()

```
#include <iostream>

using namespace std;

int main() {
    string t;
    cin >> t;
    int n;
    cin >> n;
    int sumans = 0;
    for (int test = 0; test < n; ++test) {
        string s;
        cin >> s;
        int ans = 1000000000;
        for (int i = 0; i < s.size(); ++i) {
            int pos1 = 0, pos2 = i;
            int nans = 0;
            while (pos1 < t.size()) {
                if (pos2 < s.size() && t[pos1] == s[pos2]) {
                    ++pos1;
                    ++pos2;
                } else {
                    ++nans;
                    ++pos1;
                }
            }
            ans = min(ans, nans);
        }
        //cout << ans << "\n";
        sumans += ans;
    }
    cout << sumans;
    return 0;
}
```

Task D ()

```

#include <bits/stdc++.h>
#define int long long
// #define _GLIBCXX_DEBUG

using namespace std;

int n, m;

int check(int x, int y) {
    return x >= 0 && y >= 0 && x < n && y < m;
}

int32_t main() {
    // freopen("input.txt", "r", stdin);

    cin >> n >> m;
    n += 2, m += 2;
    int sx, sy, fx, fy;
    cin >> sx >> sy >> fx >> fy;
    if (sx == fx && sy == fy) {
        cout << 0;
        return 0;
    }
    // --sx, --sy, --fx, --fy;
    vector<vector<pair<int, int>>> a(n + 2, vector<pair<int, int>> (m + 2));
    for (int i = 1; i + 1 < n; ++i) {
        for (int j = 1; j + 1 < m; ++j) {
            cin >> a[i][j].first >> a[i][j].second;
        }
    }
    vector<vector<pair<int, int>>> w(n * m);
    for (int i = 1; i + 1 < n; ++i) {
        for (int j = 1; j + 1 < m; ++j) {
            w[i * m + j].push_back(make_pair((i + a[i][j].first) * m + (j + a[i][j].second), 0));
        }
    }
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < m; ++j) {
            vector<int> dx = {-1, 1, 0, 0};
            vector<int> dy = {0, 0, -1, 1};
            for (int k = 0; k < 4; ++k) {
                int x2 = i + dx[k], y2 = j + dy[k];
                if (x2 >= 0 && x2 < n && y2 >= 0 && y2 < m) {
                    w[i * m + j].push_back(make_pair(x2 * m + y2, 1));
                }
            }
        }
    }
    vector<int> d(n * m, 1000000000000000);
    d[(sx + a[sx][sy].first) * m + sy + a[sx][sy].second] = 0;
    set<pair<int, int>> st;
    for (int i = 0; i < n * m; ++i) {
        st.insert({d[i], i});
    }
    while (st.size()) {
        int u = (*st.begin()).second;
        st.erase(st.begin());
        for (auto& vt : w[u]) {
            int v = vt.first, c = vt.second;
            if (d[v] > d[u] + c) {
                st.erase({d[v], v});
                d[v] = d[u] + c;
                st.insert({d[v], v});
            }
        }
    }
    cout << d[fx * m + fy];
    return 0;
}

```

Task E ()

```
#include <bits/stdc++.h>
#define int long long
// #define _GLIBCXX_DEBUG
typedef long double ld;

using namespace std;

int32_t main() {

    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    int n, m, B;
    cin >> n >> m >> B;
    int p, q;
    cin >> p >> q;
    --p, --q;
    int ls = 1000000;
    cout << "?_ " << -1 * ls << "_ " << -1 * ls << "_ " << ls << "_ " << ls << "\n";
    int x, y;
    cin >> x >> y;
    if (x == ls) ls *= -1;
    cout << "!_ " << ls - p << "_ " << ls - q << "\n";
    return 0;
}
```

Task F ()

```
#include <bits/stdc++.h>
// #define int long long
// #define _GLIBCXX_DEBUG
typedef long double ld;

using namespace std;

int dfs(int u, int p, vector<int>& used, vector<vector<int>>& w) {
    used[u] = 1;
    int ans = 0;
    for (int v = 0; v < used.size(); ++v) {
        if (w[u][v] == 1 && v != p) {
            if (used[v] == 1) return true;
            else if (used[v] == 0) ans |= dfs(v, u, used, w);
        }
    }
    used[u] = 2;
    return ans;
}

int dfs2(int u, vector<int>& used, vector<vector<int>>& w) {
    used[u] = 1;
    for (int v = 0; v < used.size(); ++v) {
        if (w[u][v] == 1) {
            if (used[v] == 0) dfs2(v, used, w);
        }
    }
}

int32_t main() {
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    int n, m;
    cin >> n;
    m = (n * n * n - n) / 6;
    if (n == 5) {
        cout << "0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_5_0_60_0_60\n";
        return 0;
    } else if (n == 6) {
        cout << "0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_6_0_0_120_90_0_360_360_0_0_360\n";
        return 0;
    } else if (n == 7) {
        cout << "0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_7_0_0_0_210_0_420_0_1260_0_3360_0_1470_0_5040_0_2520_0_0_0_2520\n";
        return 0;
    } else if (n == 8) {
        cout << "0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_0_8_0_0_0_0_336_0_0_840_560_3360_0_0_11760_6720_6720_5040_3360_30240_20160_0_20160_26880_20160_0_5040_20160_40320_0_20160_0_0_0_20160\n";
        return 0;
    }
    vector<int> ans(m + 1);
    for (int mask = 0; mask < (1ll << (n * (n - 1) / 2)); ++mask) {
        vector<vector<int>> w(n, vector<int>(n));
        int mask1 = mask;
        int ptrr = 0;
        for (int i = 0; i < n; ++i) {
            for (int j = i + 1; j < n; ++j) {
                w[i][j] = mask1 % 2;
                mask1 /= 2;
            }
        }
        for (int i = 0; i < n; ++i) {
            for (int j = i + 1; j < n; ++j) {
                w[j][i] = w[i][j];
            }
        }
        vector<int> used(n);
        if (dfs(0, 0, used, w)) continue;
        used.assign(n, 0);
    }
}
```

```

dfs2(0, used, w);
int fl = 1;
for (int i = 1; i < n && fl; ++i) {
    if (!used[i]) fl = 0;
}
if (fl == 0) continue;

int sum = 0;

for (int i = 0; i < n; ++i) {
    vector<int> d(n, -1);
    d[i] = 0;
    deque<int> dq;
    dq.push_back(i);
    while (dq.size()) {
        int u = dq.front();
        dq.pop_front();
        for (int v = 0; v < n; ++v) {
            if (w[u][v] == 1) {
                if (d[v] == -1) {
                    d[v] = d[u] + 1;
                    dq.push_back(v);
                }
            }
        }
    }
    for (int j = i + 1; j < n; ++j) sum += d[j];
}
++ans[sum];
}
for (int i = 1; i <= m; ++i) cout << ans[i] << " ";
return 0;
}

```