Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	63	18	421

Task A ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;

const int MAXN = 1000;
//#define int long long
const int INF = 1e9;

using itn = int;

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n;
    cin >> n;

    cout << n - 1 << '\n';
    return 0;

    if (n == 1){
        cout << 0 << '\n';
        return 0;
    }

    cout << (n + 1) / 2 << '\n';

    return 0;
}
```

Task B ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;


typedef long double ld;

const int MAXN = 1000;
#define int long long
const int INF = 1e9;
const ld EPS = 1e-3;

int eq(ld a, ld b){
    return abs(a - b) < EPS;
}

using itn = int;


struct pt {
    ld x, y;
    pt() {}
    pt(ld xx, ld yy){
        x = xx;
        y = yy;
    }
};

inline ld dist(const pt& a, const pt& b){
    return sqrtl((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));
}

struct line {
    ld a, b, c;
    line() {}
};

line to_ln(const pt& a, const pt& b){
    line l;
    l.a = b.y - a.y;
    l.b = a.x - b.x;
    l.c = -a.x * b.y + a.y * b.x;

    ld sq = sqrtl(l.a * l.a + l.b * l.b);
    l.a /= sq;
    l.b /= sq;
    l.c /= sq;

    return l;
}

ld dl(const line& l, const pt& a){
    return abs(l.a * a.x + l.b * a.y + l.c);
}

pt vct(const pt& a, const pt& b){
    pt c;
    c.x = b.x - a.x;
    c.y = b.y - a.y;

    ld sq = sqrtl(c.x * c.x + c.y * c.y);

    c.x /= sq;
    c.y /= sq;

    return c;
}
```

```
pt to(const pt& a, const pt& v){
    pt c;
    c.x = a.x + v.x;
    c.y = a.y + v.y;
    return c;
}

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);
    cout.precision(30);

    //cerr << ang << '\n';

    int n;
    cin >> n;
    vector<pt> a;
    ld x,y;
    for (int i= 0; i < n; i++){
        cin >> x >> y;
        a.eb(x, y);
    }

    if (n == 6LL){
        ld mn = (ld)1e18;

        ld mx = (ld)0;

        for (int i =0; i < n; i++){
            for (itn j = 0; j < n; j++){
                if (i == j) continue;
                ld d= dist(a[i], a[j]);
                if (d < mn){
                    mn = d;
                }
                if (d > mx){
                    mx = d;
                }
            }
        }

        int aa = -1, bb = -1, cc = -1;

        int f1 = -1;
        int f2 = -1;

        for (int i = 0; i < n; i++){
            for (int j = 0; j < n; j++){
                ld d = dist(a[i], a[j]);
                if (eq(d, mn)) f1 = j;
                if (eq(d, mx)) f2 = j;
            }
            if (f1 != -1 && f2 != -1){
                aa = i;
                bb = f2;
                cc = f1;
                break;
            }
        }

        cout <<fixed<< a[aa].x << '␣' << a[aa].y << '\n';
        cout <<fixed<< a[bb].x << '␣' << a[bb].y << '\n';
        cout <<fixed<< a[cc].x << '␣' << a[cc].y << '\n';
        return 0;
    }

    ld mn = (ld)1e18;

    ld mx = (ld)0;

    for (int i =0; i < n; i++){
        for (itn j = 0; j < n; j++){
            if (i == j) continue;
        ld d= dist(a[i], a[j]);
        if (d < mn){
            mn = d;
        }
        if (d > mx){
            mx = d;
        }
    }
}

int aa = -1, bb = -1, cc = -1;

int f1 = -1;
int f2 = -1;

for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++){
        ld d = dist(a[i], a[j]);
        if (eq(d, mn)) f1 = j;
        if (eq(d, mx)) f2 = j;
    }
    if (f1 != -1 && f2 != -1){
        aa = i;
        bb = f2;
        cc = f1;
        break;
    }
}
//cout << aa << ' ' << bb << ' ' << cc << '\n';

line l = to_ln(a[aa], a[bb]);
ld d = dist(a[aa], a[cc]);

pt norm;
norm.x = l.a;
norm.y = l.b;

vector<pt> cur;

ld dll = dl(l, a[cc]);
norm.x *= dll;
norm.y *= dll;

//cerr << l.a << ' ' << l.b << ' ' << l.c << '\n';
//cerr << norm.x << ' ' << norm.y << ' ' << dll << '\n';

pt tto1 = to(a[cc], norm);
if (eq(l.a * tto1.x + l.b * tto1.y + l.c, (ld)0)){
    tto1 = to(tto1, norm);
} else {
    norm.x *= (ld)-1;
    norm.y *= (ld)-1;
    tto1 = to(tto1, norm);
    tto1 = to(tto1, norm);
    tto1 = to(tto1, norm);
}

pt v = vct(a[aa], a[bb]);
v.x *= d;
v.y *= d;

pt tv = to(a[cc], v);

pt tto2 = to(tv, norm);
if (eq(l.a * tto2.x + l.b * tto2.y + l.c, (ld)0)){
    tto2 = to(tto2, norm);
} else {
    norm.x *= (ld)-1;
    norm.y *= (ld)-1;
    tto2 = to(tto2, norm);
    tto2 = to(tto2, norm);
    tto2 = to(tto2, norm);
}
```

```
    cur.eb(a[aa]);
    cur.eb(a[cc]);
    cur.eb(tv);
    cur.eb(a[bb]);
    cur.eb(tto2);
    cur.eb(tto1);

    for (auto i : cur){
        cout << fixed << i.x << '␣' << i.y << '\n';
    }

    return 0;
}
```

Task C ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;

const int MAXN = 1000;
//#define int long long
const int INF = 1e9;

using itn = int;

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n;
    string t;
    cin >> t;
    int N = (int)t.size();
    cin >> n;
    vector<string> ss(n);
    for (int i =0; i < n; i++) {
        cin >> ss[i];
    }

    int ans = 0;

    for (int i = 0; i < n; i++){
        int m = (int)ss[i].size();
        int cura = N;
        for (int j = 0; j < m; j++){
            int res = 0;
            int l = 0;
            while (l < N) {
                res = l;
                if (t[l] == ss[i][j]) {
                    //int l = 0;
                    int r = j;
                    //int res = 0;
                    while (l < N) {
                        if (r < m && t[l] == ss[i][r]) {
                            l++;
                            r++;
                        } else {
                            l++;
                            res++;
                        }
                    }
                    cura = min(cura, res);
                }
                l++;
            }
        }
        ans += cura;
    }

    cout << ans << '\n';

    return 0;
}
```

Task D ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;

const int MAXN = 1000;
//#define int long long
const int INF = 1e9;

using itn = int;

int d[MAXN][MAXN];
pair<int, int> g[MAXN][MAXN];

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n, m;
    cin >> n >> m;
    int ax, ay, bx, by;
    cin >> ax >> ay >> bx >> by;

    ax--; ay--; bx--; by--;

    for (itn i =0; i < n; i++){
        for (itn j = 0; j < m; j++){
            cin >> g[i][j].ff >> g[i][j].ss;
        }
    }

    set<pair<pair<int, int>, int>> q;
    memset(d, 0x3F, sizeof d);
    d[ax][ay] = 0;
    q.insert(mp(mp(ax, ay), 0));

    while (!q.empty()){
        auto res = *q.begin();
        auto pr = res.ff;
        q.erase(q.begin());

        int uu = pr.ff + g[pr.ff][pr.ss].ff;
        int vv = pr.ss + g[pr.ff][pr.ss].ss;


        for (int i = 0; i < n; i++){
            for (int j = 0; j < m; j++){
                int u = i - uu;
                int v = j - vv;
                u = abs(u);
                v = abs(v);
                if (d[i][j] > d[pr.ff][pr.ss] + u + v){
                    //if (i == bx && j == by)
                    //    cerr << pr.ff << ' ' << pr.ss << ' ' << u << ' ' << v << '\n';

                    if (q.count(mp(mp(i, j), d[i][j])))
                        q.erase(mp(mp(i, j), d[i][j]));
                    d[i][j] = d[pr.ff][pr.ss] + u + v;
                    q.insert(mp(mp(i, j), d[i][j]));
                }
            }
        }
        //break;
    }

    cout << d[bx][by] << '\n';

    return 0;
}
```

Task E ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;

const int MAXN = 300000;
#define int long long
const int INF = 1e9;

using itn = int;

struct pt {
    itn x, y;
    pt() {}
    pt(itn xx, int yy){
        x = xx;
        y = yy;
    }
};

int Q = 0;

int dead[MAXN];
int n, m;

int MX = (int)2e18;

int go(int x, int y, int cnt){
    int xx = -1e18;
    int yy = -1e18;
    int k1 = 0;
    int k2 = 0;
    int murdered = 0;
    while (cnt){
        while(dead[k1]) k1++;
        k2 = k1 + 1;
        while(k2 <= k1 || dead[k2]) k2++;

        int r1 = (m * k1) / (int)(MX);
        r1 *= n;
        int c1 = (m * k1) % (int)(MX);

        r1 += xx;
        c1 += yy;

        int r2 = (m * k2) / (int)(MX);
        r2 *= n;
        int c2 = (m * k2) % (int)(MX);

        r2 += yy;
        c2 += xx;

        Q++; assert(Q < 8191LL);
        assert((int)-1e18 <= r1+x);
        assert((int)1e18 >= r1+x);

        assert((int)-1e18 <= r2+x);
        assert((int)1e18 >= r2+x);

        assert((int)-1e18 <= c1+y);
        assert((int)1e18 >= c1+y);

        assert((int)-1e18 <= c2+y);
        assert((int)1e18 >= c2+y);

        cout << "?␣" << r1+x << '␣' << c1+y << '␣' << r2+x << '␣' << c2+y << endl;
        int rd, cd;
        cin >> rd >> cd;
        rd -= xx;
        cd -= yy;

        int kk = (rd / n) * ((int)MX / m);
        kk += (cd / m);
        //cerr << kk << '\n';

        if (!dead[kk]) murdered++;
        dead[kk] = 1;

        k1 = k2 + 1;
        cnt--;
    }
    return murdered;
}

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int b;
    cin >> n >> m >> b;
    vector<pt> a;
    itn x, y;
    for (int i = 0; i < b; i++){
        cin >> x >> y;
        a.eb(x, y);
    }

    memset(dead, 0, sizeof dead);

    for (int i = 0; i < b; i++){
        go(a[i].x-1, a[i].y-1, (1LL << (b - i - 1)));
    }

    int xx = -1e18;
    int yy = -1e18;
    for (int i = 0; i < MAXN; i++){
        if (!dead[i]){
            int r1 = (m * i) / (int)(MX);
            r1 *= n;
            int c1 = (m * i) % (int)(MX);

            r1 += xx;
            c1 += yy;

            cout << "! " << r1 << ' ' << c1 << '\n';
            return 0;
        }
    }

    return 0;
}
```

Task F ()

```
#include <bits/stdc++.h>

#define ff first
#define ss second
#define mp make_pair
#define eb emplace_back

using namespace std;

const int MAXN = 200000;
//#define int long long
const int INF = 1e9;

using itn = int;

struct Node {
    int x;
    Node() {}
};
Node t[4];

void pull(int v){
    t[v].x = t[(v << 1)].x + t[(v << 1) + 1].x;
}

void build(int v, int l, int r){
    if (l == r-1){
        t[v].x = 0;
        return;
    }
    int m = (l + r) >> 1;
    build((v << 1), l, m);
    build((v << 1) + 1, m, r);
    pull(v);
}

void add(int v, int l, int r, int pos, int x){
    if (l == r-1){
        t[v].x += x;
        return;
    }
    int m = (l + r) >> 1;
    if (pos < m){
        add(v << 1, l, m, pos, x);
    } else {
        add((v << 1) + 1, m, r, pos, x);
    }
    pull(v);
}

int get(int v, int l, int r, int tl, int tr){
    if (l >= tr || r <= tl){
        return 0;
    }
    if (l >= tl && r <= tr){
        return t[v].x;
    }
    int m = (l + r) >> 1;
    int r1 = get(v << 1, l, m, tl, tr);
    int r2 = get((v << 1) + 1, m, r, tl, tr);
    return r1 + r2;
}

#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

using namespace __gnu_pbds;

typedef tree <
        int,
        null_type,
        less<>,
        rb_tree_tag,
        tree_order_statistics_node_update
> ordered_set;

int ans[MAXN];
vector<pair<int, int>> e;
int d[12][12];

void gen(int c, int n){
    if (c == n-1){
        memset(d, 0x3F, sizeof d);
        for (auto i : e) {
            d[i.ff][i.ss] = 1;
            d[i.ss][i.ff] = 1;
            //cerr << i.ff << ' ' << i.ss << '\n';
        }
        for (int k = 0; k < n; k++){
            for (int i = 0; i < n; i++){
                for (int j = 0; j < n; j++){
                  d[i][j] = min(d[i][j], d[i][k] + d[k][j]);
                }
            }
        }
        int res = 0;
        for (int i = 0; i < n; i++){
            for (int j = i + 1; j < n; j++){
                if (d[i][j] > 100) return;
                res += d[i][j];
            }
        }
        ans[res]++;
        return;
    }
    for (int i =0; i < n; i++){
        for (int j = i + 1; j < n; j++){
            if ((int)e.size() > 0 && mp(i, j) <= e.back()) continue;
            e.eb(i, j);
            gen(c+1, n);
            e.pop_back();
        }
    }
}

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0);

    int n;
    cin >> n;
    itn m;
    cin >> m;

    /*
    if (n == 1){
        cout << "1\n";
        return 0;
    }
    if (n == 3){
        cout << "0 0 0 3\n";
        return 0;
    }
    if (n == 4){
        cout << "0 0 0 0 0 0 0 0 4 12\n";
        return 0;
    }
     */
    memset(ans, 0, sizeof ans);
    gen(0, n);
    for (int i = 1; i < m+1; i++){
        cout << ans[i] << '␣';
    }
    cout << '\n';

    return 0;
}
```