Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	40	100	27	467

Task A ()

```
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pll pair<ll, ll>
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

int main() {
    fast;
    ll a; cin>>a;
    cout<<a-1;
}
```

Task B ()

```
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pb push_back
#define pll pair<ll, ll>
#define pdd pair<ld, ld>
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

int a;

void solve1(){
    vec<pdd> m(a);
    for(int q=0; q<a; q++) cin>>m[q].F>>m[q].S;
    ld x1 = m[0].F, y1 = m[0].S;
    ld x2, y2, mxd = 0;
    for(int q=1; q<6; q++){
        ld D = sqrtl(pow(m[0].F-m[q].F, 2) + pow(m[0].S-m[q].S, 2));
        if(D > mxd) mxd = D, x2 = m[q].F, y2 = m[q].S;
    }
    ld mx = (x1+x2)/2, my = (y1+y2)/2;
    cout<<mx<<" "<<my<<endl;
    cout<<m[0].F<<" "<<m[0].S<<endl;
    cout<<mxd/2<<" "<<mxd/2<<endl;
}

const ld PI = 3.14159265358979323846264338327950;
void solve2(){
    ld cx, cy; cin>>cx>>cy;
    ld fx, fy; cin>>fx>>fy;
    vec<pdd> o;
    ld dx = cx, dy = cy;
    cx = cy = 0;
    fx -= dx, fy -= dy;
    ld r; cin>>r;
    ld ang = atan2(fy, fx);
    for(int q=0; q<6; q++){
        ld nx = r*cos(ang+PI/3*q), ny = r*sin(ang+PI/3*q);
        o.pb({nx, ny});
    }
    for(pdd p : o) cout<<p.F+dx<<" "<<p.S+dy<<endl;
}

int main() {
    fast;
    cout.precision(12);
    cout<<fixed;
    cin>>a;
    if(a==6) solve1();
    else solve2();
}
```

Task C ()

```
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pb push_back
#define pll pair<ll, ll>
#define pdd pair<ld, ld>
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

int a,b,z;
str t;

int solve(str &u){
    a = t.size();
    b = u.size();
    int mx = 0;
    for(int st=0; st<b; st++){
        int tyt = 0, i = st, j = 0;
        for(; i<b && j<a; j++){
            if(t[j] == u[i]) i++, tyt++;
        }
        mx = max(mx, tyt);
    }
    return a-mx;
}

int main() {
    fast;
    cin>>t>>z;
    int o = 0;
    for(int q=0; q<z; q++){
        str u; cin>>u;
        o += solve(u);
    }
    cout<<o;
}
```

Task D ()

```cpp
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pb push_back
#define pll pair<ll, ll>
#define pq priority_queue
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

struct cmp{
    bool operator()(const pii &p1, const pii &p2){
        return p1.S > p2.S;
    }
};

int a,b,inf=1000000000;
vec<vec<int>> dx, dy;
vec<vec<pii>> l;
int sx,sy,fx,fy;

int main() {
    fast;
    cin>>a>>b;
    cin>>sx>>sy>>fx>>fy;
    sx--, sy--, fx--, fy--;
    dx = vec<vec<int>>(a, vec<int>(b));
    dy = vec<vec<int>>(a, vec<int>(b));
    l = vec<vec<pii>>(a*b, vec<pii>());
    for(int q=0; q<a; q++){
        for(int w=0; w<b; w++){
            int x,y; cin>>x>>y;
            dx[q][w] = x, dy[q][w] = y;
        }
    }
    for(int q=0; q<a; q++){
        for(int w=0; w<b; w++){
            for(int e=0; e<a; e++){
                for(int r=0; r<b; r++){
                    int v1 = q*b+w, v2 = e*b+r;
                    if(v1==v2) continue;
                    int d = abs(e-(q+dx[q][w])) + abs(r-(w+dy[q][w]));
                    l[v1].pb({v2, d});
                }
            }
        }
    }
    vec<int> dst(a*b, inf);
    int sv = sx*b+sy, fv = fx*b+fy;
    dst[sv] = 0;
    pq<pii, vec<pii>, cmp> PQ;
    PQ.push({sv, 0});
    for(; PQ.size();){
        pii t = PQ.top(); PQ.pop();
        int v = t.F, d = t.S;
        if(d != dst[v]) continue;
        for(pii p : l[v]){
            if(d+p.S < dst[p.F]){
                dst[p.F] = d+p.S;
                PQ.push({p.F, dst[p.F]});
            }
        }
    }
    //for(int q=0; q<a*b; q++) cout<<q<<": "<<dst[q]<<endl;
    cout<<dst[fv];
}
```

Task E ()

```
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pb push_back
#define pll pair<ll, ll>
#define pdd pair<ld, ld>
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

ll a,b,k;
vec<pll> m;
vec<ll> nums;

int main() {
    fast;
    cin>>a>>b>>k;
    ll all = 1<<k;
    m = vec<pll>(k);
    for(int q=0; q<all*2; q++) nums.pb(q);
    for(int q=0; q<k; q++){
        ll x,y; cin>>x>>y; x--, y--;
        m[q] = {x,y};
    }
    //if(k>1) assert(0);
    set<ll> rem;
    set<pll> XYU;
    for(int it=0; it<k; it++){
        ll stp = 1<<(k-it-1), i=0;
        for(int q=0; q<stp; q++){
            ll n1=-1, n2=-1;
            for(; rem.count(nums[i]);) i++;
            n1 = nums[i++];
            for(; rem.count(nums[i]);) i++;
            //assert(i<nums.size());
            n2 = nums[i++];
            assert(rem.count(n1)+rem.count(n2)==0);

            ll x = m[it].F, y1 = n1*b+m[it].S, y2 = n2*b+m[it].S;
            //assert(XYU.count({x, y1})==0);
            //assert(XYU.count({x, y2})==0);
            cout<<"?␣"<<x<<"␣"<<y1<<"␣"<<x<<"␣"<<y2<<endl;
            XYU.insert({x, y1});
            XYU.insert({x, y2});
            ll X,Y; cin>>X>>Y;
            rem.insert(Y/b);
        }
        vec<ll> nw_num;
        for(ll i : nums) if(!rem.count(i)) nw_num.pb(i);
        nums = nw_num;
    }
    assert(nums.size()>0);
    cout<<"!␣0␣"<<nums[0]*b<<endl;
}
```

Task F ()

```
#pragma GCC optimize("Ofast")
//#pragma GCC target("avx,avx2,fma")
#include <bits/stdc++.h>
#define F first
#define S second
#define vec vector
#define pll pair<ll, ll>
#define pii pair<int, int>
#define uset unordered_set
#define umap unordered_map
#define pnn pair<Node*, Node*>
#define uid uniform_int_distribution
#define fast cin.tie(0); cout.tie(0); cin.sync_with_stdio(0); cout.sync_with_stdio(0);
using namespace std;
typedef string str;
typedef long long ll;
typedef long double ld;
mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

vec<str> o = {
        "1",
        "0␣0␣0␣3",
        "0␣0␣0␣0␣0␣0␣0␣0␣4␣12",
        "0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣5␣0␣60␣0␣60",
        "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",
        "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",
        "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␣0␣20160␣0␣0␣0␣0␣20160",
        "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␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣0␣9␣0␣0␣0␣0␣504␣0␣0␣0␣1512␣0␣10080␣0␣0␣0␣33264␣0␣
            75600␣0␣22680␣0␣189000␣0␣231840␣0␣136080␣0␣529200␣0␣241920␣0␣362880␣0␣665280␣0␣181440␣
            0␣544320␣0␣423360␣0␣0␣0␣408240␣0␣362880␣0␣0␣0␣181440␣0␣0␣0␣0␣0␣181440",
        "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␣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␣10␣0␣0␣0␣0␣0␣0␣
            720␣0␣0␣0␣0␣2520␣0␣15120␣5040␣3150␣0␣0␣80640␣0␣100800␣131040␣75600␣75600␣0␣604800␣
            166320␣226800␣907200␣579600␣781200␣0␣1209600␣1844640␣1058400␣1890000␣1512000␣2570400␣
            453600␣1058400␣5745600␣3024000␣831600␣2268000␣3931200␣2772000␣1814400␣5443200␣4233600␣
            4384800␣0␣1814400␣6048000␣2268000␣907200␣3628800␣6955200␣302400␣0␣5443200␣1814400␣
            3628800␣0␣1814400␣2419200␣0␣0␣1814400␣3628800␣453600␣0␣3628800␣0␣0␣0␣0␣1814400␣0␣0␣0␣0
            ␣0␣0␣1814400"
};

int main() {
    fast;
    int x; cin>>x;
    cout<<o[x-2];
}
```