Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	45	0	445

Task A ()

```cpp
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define F first
#define S second
#define PB emplace_back
#define rs resize

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ll mod = 1e9 + 7;
const ld eps = 1e-9;

ll rev(ll a, ll m = mod){
    return a == 1? 1 : m - m * rev(m % a,a) / a;
}


int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    ll n; cin>>n;
    cout<<n - 1<<endl;
    return 0;
}
```

Task B ()

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define F first
#define S second
#define PB emplace_back
#define rs resize

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ll mod = 1e9 + 7;
const ld eps = 1e-2;

struct Point{
    ld x, y;
};

ostream& operator<<(ostream& out,Point a){
    out<<a.x<<'␣'<<a.y;
    return out;
}

bool operator==(Point a,Point b){
    return (fabs(a.x - b.x) <= eps && fabs(a.y - b.y) <= eps);
}

Point operator+(Point a,Point b){
    Point res = {a.x + b.x, a.y + b.y};
    return res;
}

Point operator-(Point &a,Point &b){
    Point res = {a.x - b.x, a.y - b.y};
    return res;
}

Point operator/(Point a,ld k){
    Point res = {a.x / k, a.y / k};
    return res;
}

ll rev(ll a, ll m = mod){
    return a == 1? 1 : m - m * rev(m % a,a) / a;
}


int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    cout<<fixed<<setprecision(20);
    //cout<<(3.0L * sqrt(3) + 1) / 2<<endl;
    int n; cin>>n;
    vector <Point> v(n); for(auto &e:v) cin>>e.x>>e.y;
    if(n == 6){
        Point c;
        bool ok = false;
        for(int i=0;i<6;i++){
            for(int j=0;j<6;j++){
                for(int k=0;k<6;k++){
                    for(int h = 0;h<6;h++){
                        if(i == j || i == k || i == h || j == k || j == h || h == k) continue;
                        if((v[i] + v[j]) == (v[k] + v[h])){
                            c = (v[i] + v[j]) / 2.0L;
                            v[0] = v[i];
                            ok = true;
                            break;
                        }
                    }if(ok) break;
                }if(ok) break;
            }if(ok) break;
        }
        assert(ok);
        cout<<v[0]<<endl<<c<<endl<<c<<endl;
    }else if(n == 3){
        if(v[1] == v[2]) swap(v[1],v[0]);
        if(v[0] == v[1]) swap(v[1],v[2]);
        Point c = v[1] - v[0];
        cout<<c + v[0]<<endl;
        for(int i=1;i<6;i++){
            c = {(c.x - sqrt(3.0L) * c.y) / 2.0L, (c.y + sqrt(3.0L) * c.x) / 2.0L};
            cout<<c + v[0]<<endl;
        }
    } // 1.73205080756887719318
    return 0;
}
```

Task C ()

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define sz(v) (int)v.size()
#define F first
#define S second
#define PB emplace_back
#define rs resize

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ld eps = 1e-9;


int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    string t; cin>>t;
    vector <vector <int>> v(sz(t) + 1,vector <int>(27,-1));
    for(int i=0;i<27;i++){
        for(int j=sz(t) - 1;j>=0;j--){
            v[j][i] = (t[j] - 'a' == i?j+1:v[j+1][i]);
            //cout<<char(i + 'a')<<' '<<j<<' '<<v[j][i]<<endl;
        }
    }

    int ans = 0;
    int n; cin>>n;
    while(n--){
        string s; cin>>s;
        int mn = sz(t);
        for(int i=0;i<sz(s);i++){
            int cur = 0;
            int k = 0;
            for(int j=i;j<min(i + sz(t),sz(s));j++){
                if(v[k][s[j] - 'a'] == -1){
                    //cout<<j<<' '<<k<<' '<<sz(t) - k<<endl;
                    break;
                }else{
                    cur += v[k][s[j] - 'a'] - k - 1;
                    k = v[k][s[j] - 'a'];
                }
            }cur += sz(t) - k;
            mn = min(mn,cur);
        }
        //cout<<s<<' '<<mn<<endl;
        ans += mn;
    }cout<<ans<<endl;

    return 0;
}
```

Task D ()

```cpp
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define F first
#define S second
#define PB emplace_back
#define rs resize
#define mt make_tuple

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ld eps = 1e-9;

struct p{
    int x, y;
};

bool operator==(p a,p b){
    return a.x == b.x && a.y == b.y;
}

bool ar[1050][1050];
p c[1050][1050];
int d[1050][1050];

int dd[4][2] = {{0,1},{0,-1},{1,0},{-1,0}};

int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    for(auto &ee:d) for(auto &e:ee) e = inf;
    int n,m; cin>>n>>m;
    p f,s; cin>>f.x>>f.y>>s.x>>s.y;
    if(f == s){
        cout<<0;
        return 0;
    }
    for(int i=1;i<=n;i++) for(int j=1;j<=m;j++){
        ar[i][j] = true;
        cin>>c[i][j].x>>c[i][j].y;
    }

    priority_queue <tuple <int,int,int,int>> q; //dis,type,x,y
    p cur = f;
    int cost = abs(c[cur.x][cur.y].x) + abs(c[cur.x][cur.y].y);

    d[cur.x][cur.y] = cost;
    q.push(mt(-cost,0,cur.x,cur.y));

    int tx = cur.x + c[cur.x][cur.y].x, ty = cur.y + c[cur.x][cur.y].y;
    if(ar[tx][ty] && d[tx][ty] > 0){
        d[tx][ty] = 0;
        q.push(mt(-d[tx][ty],0,tx,ty));
    }
    for(auto e:dd){
        tx = cur.x + e[0], ty = cur.y + e[1];
        cost = abs(c[cur.x][cur.y].x - e[0]) + abs(c[cur.x][cur.y].y - e[1]);
        if(ar[tx][ty] && d[tx][ty] > cost){
            d[tx][ty] = cost;
            q.push(mt(-d[tx][ty],1,tx,ty));
        }
    }
    while(!q.empty()){
        int a,t; tie(a,t,cur.x,cur.y) = q.top(); q.pop(); a = -a;
        if(cur == s) break;
        //cout<<cur.x<<' '<<cur.y<<' '<<a<<' '<<t<<endl;
        /*if(t == 0){

            q.push(mt(-a - 1,1,cur.x,cur.y));
            int tx = cur.x + c[cur.x][cur.y].x, ty = cur.y + c[cur.x][cur.y].y;
            cout<<tx<<' '<<ty<<endl;
            if(ar[tx][ty] && d[tx][ty] > a){
                d[tx][ty] = a;
                q.push(mt(-d[tx][ty],0,tx,ty));
            }
        }else*/
            int tx = cur.x + c[cur.x][cur.y].x, ty = cur.y + c[cur.x][cur.y].y;
            if(ar[tx][ty] && d[tx][ty] > a){
                d[tx][ty] = a;
                q.push(mt(-d[tx][ty],0,tx,ty));
            }
            for(auto e:dd){
                tx = cur.x + e[0], ty = cur.y + e[1];
                if(ar[tx][ty] && d[tx][ty] > a + 1){
                    d[tx][ty] = a + 1;
                    q.push(mt(-d[tx][ty],1,tx,ty));
                }
            }


    }cout<<d[s.x][s.y]<<endl;

    return 0;
}

/*

1 6
1 4 1 6
0 1 0 -1 0 1 0 -1 0 -1 0 1

1 8
1 2 1 5
0 1 0 0 0 1 0 1 0 0 0 1 0 -1 0 0


1 6
1 3 1 6
1 0 -1 1 1 -1 0 1 1 1 -1 1

*/
```

Task E ()

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define F first
#define S second
#define PB emplace_back
#define rs resize

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ld eps = 1e-9;

int T = 1e4;
const ll pred = 1e18;

map <pair <ll,ll>,int> cmp;
vector <bool> al;
int cnt = 0;
void qer(ll a,ll b,ll c,ll d,int i,int j){
    assert(++cnt != 8191);
    cmp[{a,b}] = i; cmp[{c,d}] = j;
    cout<<"?␣"<<a<<'␣'<<b<<'␣'<<c<<'␣'<<d<<endl;
    cin>>a>>b;
    al[cmp[{a,b}]] = false;
}


int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    ll n,m; cin>>n>>m;
    ll x = -pred,y = 0;
    int B; cin>>B; T = (1<<B);
    vector <pair <ll,ll>> need(B);
    for(int i=0;i<B;i++){
        ll a,b; cin>>a>>b; a--, b--;
        need[i] = {a,b};
    }
    vector <pair <ll,ll>> v(T);
    al.rs(T); // alive
    for(int i=0;i<T;i++){
        v[i] = {x,y}; y += m + 1;
        if(y + m + 100 >= pred){
            y = 0, x += n + 1;
        }
        al[i] = true;
    }

    for(int i=0;i<B;i++){
        int j = -1,k = -1;
        for(int h=0;h<T;h++){
            if(!al[h]) continue;
            k = h; if(j == -1) swap(j,k);
            if(k != -1){
                qer(v[j].F + need[i].F,v[j].S + need[i].S,v[k].F + need[i].F,v[k].S + need[i].S,j,
                    k);
                k = j = -1;

            }
        }
    }
    for(int i=0;i<T;i++){
        if(al[i]){
            cout<<"!␣"<<v[i].F<<'␣'<<v[i].S<<endl;
            return 0;
        }
    }
    assert(0);
    return 0;
}
```

Task F ()

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
using ld = long double;

#define all(v) v.begin(),v.end()
#define rall(v) v.rbegin(),v.rend()
#define F first
#define S second
#define PB emplace_back
#define rs resize

const int N = 2e5 + 500;
const ll inf = INT_MAX >> 1;
const ll mod = 1e9 + 7;
const ld eps = 1e-2;

struct Point{
    ld x, y;
};

ostream& operator<<(ostream& out,Point a){
    out<<a.x<<' '<<a.y;
    return out;
}

bool operator==(Point a,Point b){
    return (fabs(a.x - b.x) <= eps && fabs(a.y - b.y) <= eps);
}

Point operator+(Point a,Point b){
    Point res = {a.x + b.x, a.y + b.y};
    return res;
}

Point operator-(Point &a,Point &b){
    Point res = {a.x - b.x, a.y - b.y};
    return res;
}

Point operator/(Point a,ld k){
    Point res = {a.x / k, a.y / k};
    return res;
}

ll rev(ll a, ll m = mod){
    return a == 1? 1 : m - m * rev(m % a,a) / a;
}


int main(){
    ios_base::sync_with_stdio(0); cin.tie(0);
    cout<<fixed<<setprecision(20);
    //cout<<(3.0L * sqrt(3) + 1) / 2<<endl;
    int n,m; cin>>n>>m;
    if(n == 3) cout<<"0 0 0 3";
    else if(n == 4) cout<<"0 0 0 0 0 0 0 0 4 12";
    else cout<<1;
    return 0;
}
```