Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	20	0	0	320

Task A ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

#define int long long       //!!!!
#define ff first
#define ss second
#define eb emplace_back
#define pb pop_back
#define sz(x) (int)x.size()
#define ld long double
#define PI acos(-1)

using namespace std;
using namespace __gnu_pbds;

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

const int INF = 1e9 + 7;
const ld EPS = 1e-6;
const int BS = 255;



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

    int n;
    cin >> n;
    cout << n - 1;

    return 0;
}
```

Task B ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

#define int long long       //!!!!
#define ff first
#define ss second
#define eb emplace_back
#define pb pop_back
#define sz(x) (int)x.size()
#define PI acos(-1)

using namespace std;
using namespace __gnu_pbds;

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

const int INF = 1e9 + 7;
const ld EPS = 1e-6;
const int BS = 255;

struct Gpoint{
    ld x, y;
    Gpoint(ld a, ld b) {
        x = a;
        y = b;
    };
    Gpoint(){};
};

struct Gvector{
    ld x, y;
    Gvector(Gpoint a, Gpoint b) {
        x = (b.x - a.x), y = (b.y - a.y);
    };
    Gvector(){};
    ld len(){
        return sqrt(x * x + y * y);
    }
};

struct Gline{
    ld a, b, c;
    Gline(Gpoint f, Gpoint s) {
        a = f.y - s.y;
        b = s.x - f.x;
        c = -a * f.x - b * f.y;
    };
};

bool cmp(Gvector a, Gvector b){
    ld cross = a.x * b.y - a.y * b.x;
    if (abs(cross) < EPS)
        return a.len() < b.len();
    return cross > 0;
}

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

    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    int n;
```

```
    cin >> n;
    cout << fixed << setprecision(10);
    if (n == 6){
        Gpoint p[6];
        for (int i = 0; i < n; i++)
            cin >> p[i].x >> p[i].y;

        Gpoint m(0, 1e5);
        for (int i = 0; i < n; i++)
            if (m.y > p[i].y || m.y == p[i].y && m.x > p[i].x)
                m = p[i];
        Gvector v[6];
        for (int i = 0; i < n; i++)
            v[i] = Gvector(m, p[i]);
        sort(v, v + n + 1, cmp);
        for (int i = 0; i < n; i++)
            p[i].x = v[i].x + m.x, p[i].y = v[i].y + m.y;

        //for (int i = 0; i < n; i++)
        //    cout << v[i].x << ' ' << v[i].y << '/' << p[i].x << ' ' << p[i].y << '\n';
        //cout << '\n';

        cout << p[0].x << '␣' << p[0].y << '\n';
        cout << p[1].x << '␣' << p[1].y << '\n';
        Gline l1(p[0], p[3]), l2(p[1], p[4]);
        ld delta = l1.a * l2.b - l2.a * l1.b;
        ld delta1 = l2.a * l1.c - l1.a * l2.c;
        ld delta2 = l1.b * l2.c - l2.b * l1.c;
        cout << delta2 / delta << '␣' << delta1 / delta << '\n';
    }
    else {
        Gpoint a, b, cen;
        cin >> a.x >> a.y >> b.x >> b.y >> cen.x >> cen.y;
        Gvector fv(cen, a), sv(cen, b);
        cout << a.x << '␣' << a.y << '\n';
        cout << b.x << '␣' << b.y << '\n';
        Gvector tv;
        tv.x = sv.x - fv.x, tv.y = sv.y - fv.y;
        Gpoint c(tv.x + cen.x, tv.y + cen.y);
        cout << c.x << '␣' << c.y << '\n';
        Gpoint d(-fv.x + cen.x, -fv.y + cen.y);
        cout << d.x << '␣' << d.y << '\n';
        Gpoint e(-sv.x + cen.x, -sv.y + cen.y);
        cout << e.x << '␣' << e.y << '\n';
        Gpoint f(-tv.x + cen.x, -tv.y + cen.y);
        cout << f.x << '␣' << f.y << '\n';
    }
    return 0;
}
```

Task C ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

#define int long long       //!!!!
#define ff first
#define ss second
#define eb emplace_back
#define pb pop_back
#define sz(x) (int)x.size()
#define PI acos(-1)

using namespace std;
using namespace __gnu_pbds;

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

const int INF = 1e9 + 7;
const ld EPS = 1e-6;
const int BS = 255;

pair < int, int > p[700], cell[700];
map < int, int > cs;
bool used[700], shot[700];

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


    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    string t, s1;
    int n, sum = 0;
    cin >> t >> n;// >> s1;
    for (int k = 0; k < n; k++){
        cin >> s1;
        int mr = INF;
        for (int i = 0; i < sz(s1); i++){
            int i1 = i, res = 0;
            for (int j = 0; j < sz(t); j++){
                if (i1 < sz(s1) && t[j] == s1[i1]){
                    i1++;
                    continue;
                }
                res++;
            }
            mr = min(res, mr);
        }
        //cout << mr << '\n';
        sum += mr;
    }
    cout << sum << '\n';
    return 0;
}
```

Task D ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>

#define int long long       //!!!!
#define ff first
#define ss second
#define eb emplace_back
#define pb pop_back
#define sz(x) (int)x.size()
#define PI acos(-1)

using namespace std;
using namespace __gnu_pbds;

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

const int INF = 1e9 + 7;
const ld EPS = 1e-6;
const int BS = 255;

int r[1010][1010], c[1010][1010], g[1010][1010], d[1000010];
bool used[1000010];

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

    //freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);

    int n, m;
    cin >> n >> m;
    int ar, ac, br, bc;
    cin >> ar >> ac >> br >> bc;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            cin >> r[i][j] >> c[i][j];
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
            for (int i1 = 0; i1 < n; i1++)
                for (int j1 = 0; j1 < m; j1++){
                    int v = abs(i - i1 + r[i][j]) + abs(j - j1 + c[i][j]);
                    g[i * m + j][i1 * m + j1] = v;
                }
    set < int > s;
    ar--, ac--, br--, bc--;
    for (int i = 0; i < n * m; i++)
        d[i] = INF;
    d[ar * m + ac] = 0;
    for (int i = 0; i < n * m; i++){
        int v = -1;
        for (int j = 0; j < n * m; j++)
            if ((v == -1 || d[j] < d[v]) && !used[j])
                v = j;
        used[v] = 1;
        for (int j = 0; j < n * m; j++)
            d[j] = min(d[j], d[v] + g[v][j]);
    }
    cout << d[br * m + bc];
    return 0;
}
```

Task E ()

Task F ()