# Олимпиада СПбГУ по информатике 2019/20 учебного года

| A | B | C | D | E | F | Sum |
|---|---|---|---|---|---|---|
| 100 | 100 | 100 | 40 | 6 | 0 | 346 |

## Task A ()

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

#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define sz(x) (long long)((x).size())
#define ll long long
#define ull unsigned long long
#define ld long double

#pragma GCC optimize ("unroll-loops")
#pragma GCC optimize "-O3"
#pragma GCC optimize ("Ofast")
#pragma GCC optimize ("fast-math")


using namespace std;

mt19937_64 rnd(time(0));

int main()
{
   // freopen("", "r", stdin);
   // freopen("", "w", stdout);
    ll n;
    cin >> n;
    cout << n - 1;
    return 0;
}
```

## Task B ()

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

#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define sz(x) (long long)((x).size())
#define ll long long
#define ull unsigned long long
typedef long double ld;

#pragma GCC optimize ("unroll-loops")
#pragma GCC optimize "-O3"
#pragma GCC optimize ("Ofast")
#pragma GCC optimize ("fast-math")


using namespace std;

mt19937_64 rnd(time(0));
struct Point{
    ld x, y;
    Point() : x(0), y(0) {}
    Point(ld x, ld y) : x(x), y(y) {}
};
ld ras(Point a, Point b)
{
    return sqrt((a.x - b.x) * (a.x - b.x) + (a.y - b.y) * (a.y - b.y));
}
ld cross_prod(Point a, Point b)
{
    return (a.x * b.y - a.y * b.x);
}
ld dot_prod(Point a, Point b)
{
    return (a.x * b.x + a.y * b.y);
}
bool comp(Point a, Point b)
{
    if (a.y == b.y)
        return a.x < b.x;
    return a.y < b.y;
}
ld lol1, lol2;
bool comp1(Point a, Point b)
{
    Point c = Point(a.x - lol1, a.y - lol2);
    Point d = Point(b.x - lol1, b.y - lol2);
    return atan2(c.y, c.x) < atan2(d.y, d.x);
}
ld eps = 1e-7;
int main()
{
    ll n;
    cin >> n;
    cout << fixed << setprecision(10);
    if (n == 6)
    {
        vector <Point> mas(n);
        for (int i = 0; i < n; i++)
        {
            ld a, b;
            cin >> a >> b;
            mas[i] = Point(a, b);
        }
        sort(all(mas), comp);
        lol1 = mas[0].x;
        lol2 = mas[0].y;
        sort(all(mas), comp1);
        for (int i = 0; i < 3; i++)
            cout << mas[i].x << "␣" << mas[i].y << endl;
        return 0;
    }
    vector <Point> mas(6);
    for (int i = 0; i < n; i++)
        cin >> mas[i].x >> mas[i].y;
    for (int i = 0; i < 3; i++)
        cout << mas[i].x << "␣" << mas[i].y << endl;
    ld dist = ras(mas[0], mas[2]);
    ld st = ras(mas[0], mas[1]);
    swap(st, dist);
    for (int i = 3; i < 6; i++)
    {
        Point a = Point(mas[i - 3].x - mas[i - 1].x, mas[i - 3].y - mas[i - 1].y);
        Point vec1 = Point(a.y, a.x);
        Point vec2 = Point(-a.y, a.x);
        Point vec3 = Point(a.y, -a.x);
        Point vec4 = Point(-a.y, -a.x);
        Point vecfirst = Point(mas[i - 3].x - mas[i - 2].x, mas[i - 3].y - mas[i - 2].y);
        Point vecsecond = Point(mas[i - 2].x - mas[i - 1].x, mas[i - 2].y - mas[i - 1].y);
        if (fabs(dot_prod(vec1, a)) <= eps)
        {
            Point next = mas[i - 1];
            vec1.x = vec1.x / st * dist;
            vec1.y = vec1.y / st * dist;
            next.x += vec1.x;
            next.y += vec1.y;
            Point vecthird = Point(mas[i - 1].x - next.x, mas[i - 1].y - next.y);
            if (cross_prod(vecfirst, vecsecond) * cross_prod(vecsecond, vecthird) >= 0)
            {
                cout << next.x << "␣" << next.y << endl;
                mas[i] = next;
                continue;
            }
        }
        if (fabs(dot_prod(vec2, a)) <= eps)
        {
            Point next = mas[i - 1];
            vec2.x = vec2.x / st * dist;
            vec2.y = vec2.y / st * dist;
            next.x += vec2.x;
            next.y += vec2.y;
            Point vecthird = Point(mas[i - 1].x - next.x, mas[i - 1].y - next.y);
            if (cross_prod(vecfirst, vecsecond) * cross_prod(vecsecond, vecthird) >= 0)
            {
                cout << next.x << "␣" << next.y << endl;
                mas[i] = next;
                continue;
            }
        }
        if (fabs(dot_prod(vec3, a)) <= eps)
        {
            Point next = mas[i - 1];
            vec3.x = vec3.x / st * dist;
            vec3.y = vec3.y / st * dist;
            next.x += vec3.x;
            next.y += vec3.y;
            Point vecthird = Point(mas[i - 1].x - next.x, mas[i - 1].y - next.y);
            if (cross_prod(vecfirst, vecsecond) * cross_prod(vecsecond, vecthird) >= 0)
            {
                cout << next.x << "␣" << next.y << endl;
                mas[i] = next;
                continue;
            }
        }
        if (fabs(dot_prod(vec4, a)) <= eps)
        {
            Point next = mas[i - 1];
            vec4.x = vec4.x / st * dist;
            vec4.y = vec4.y / st * dist;
            next.x += vec4.x;
            next.y += vec4.y;
            Point vecthird = Point(mas[i - 1].x - next.x, mas[i - 1].y - next.y);
            if (cross_prod(vecfirst, vecsecond) * cross_prod(vecsecond, vecthird) >= 0)
            {
                cout << next.x << "␣" << next.y << endl;
                mas[i] = next;
                continue;
            }
        }
    }
    return 0;
}
```

## Task C ()

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

#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define sz(x) (long long)((x).size())
typedef long long ll;
typedef unsigned long long ull;
typedef long double ld;

#pragma GCC optimize ("unroll-loops")
#pragma GCC optimize "-O3"
#pragma GCC optimize ("Ofast")
#pragma GCC optimize ("fast-math")


using namespace std;

mt19937_64 rnd(time(0));

int main()
{
   // freopen("", "r", stdin);
   // freopen("", "w", stdout);
   string s;
   cin >> s;
   ll n;
   cin >> n;
   ll oans = 0;
   for (int i = 0; i < n; i++)
   {
        ll ans = sz(s);
        string lol;
        cin >> lol;
        for (int from = 0; from < sz(lol); from++)
        {
            ll l = 0, cnt = sz(s);
            for (int j = from; j < sz(lol); j++)
            {
                if (l >= sz(s))
                {
                    break;
                }
                while (lol[j] != s[l] && l < sz(s))
                {
                    l++;
                }
                if (l < sz(s))
                    cnt--;
              //  cout << l << " " << j << " " << cnt << endl;
                l++;
            }
          //  cout << endl;
            ans = min(ans, cnt);
        }
       // cout << ans << endl;
        oans += ans;
   }
   cout << oans << endl;
    return 0;
}
```

## Task D ()

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

#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define sz(x) (long long)((x).size())
#define ll long long
#define ull unsigned long long
typedef long double ld;

/*
#pragma GCC optimize ("unroll-loops")
#pragma GCC optimize "-O3"
#pragma GCC optimize ("Ofast")
#pragma GCC optimize ("fast-math")
*/

using namespace std;

mt19937_64 rnd(time(0));
signed main()
{
    int n, m;
    cin >> n >> m;
    pair <int, int> a[n][m];
    int a1, b, c, d;
    cin >> a1 >> b >> c >> d;
    a1--;
    b--;
    c--;
    d--;
    int from = a1 * m + b;
    int to = c * m + d;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < m; j++)
        {
            cin >> a[i][j].first >> a[i][j].second;
        }
    vector <bool> used(n * m, false);
    vector <int> dist(n * m, INT_MAX);
    dist[from] = 0;
    for (int l = 0; l < m * n; l++)
    {
        int v = -1;
        for (int now = 0; now < n * m; now ++)
        {
            if (used[now]) continue;
            if (v == -1 || (dist[now] < dist[v]))
                v = now;
        }
        if (v == -1) break;
        used[v] = true;
        int i = v / m;
        int j = v - m * i;
        for (int i1 = 0; i1 < n; i1++)
        {
            for (int j1 = 0; j1 < m; j1++)
            {
                int num = i1 * m + j1;
                int delta = abs(-i1 + i + a[i][j].first) + abs(-j1 + j + a[i][j].second);
                dist[num] = min((ll)dist[num], (ll)dist[v] + (ll)delta);
            }
        }
    }
    cout << dist[to];
    return 0;
}
```

## Task E ()

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

#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define sz(x) (long long)((x).size())
#define ll long long
#define ull unsigned long long
typedef long double ld;

/*
#pragma GCC optimize ("unroll-loops")
#pragma GCC optimize "-O3"
#pragma GCC optimize ("Ofast")
#pragma GCC optimize ("fast-math")
*/

using namespace std;

mt19937_64 rnd(time(0));
signed main()
{
    int n, m, B;
    cin >> n >> m >> B;
    vector <pair <int, int> > mas(2);
    for (int i = 0; i < B; i++)
    {
        cin >> mas[i].first >> mas[i].second;
    }
    m--;
    n--;
    if (B == 1)
    {
        while (true)
        {
            cout << "?␣" << mas[0].first << "␣" << mas[0].second << "␣" << mas[0].first + 3000000
                << "␣" << mas[0].second + 3000000 << endl;
            int a, b;
            cin >> a >> b;
            if (a > 300000 && b > 300000)
            {
                cout << '!';
                cout << "␣";
                cout << 1 << "␣" << 1 << endl;
                return 0;
            }
            else
            {
                cout << '!';
                cout << "␣";
                cout << 3000001 << "␣" << 3000001 << endl;
                return 0;
            }
        }
    }
    return 0;
}
```

**Task F ()**