Олимпиада СПбГУ по информатике 2019/20 учебного года

A	B	C	D	E	F	Sum
100	100	100	0	100	0	400

Task A ()

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

#define fi first
#define se second
#define sz(x) (int)(x).size()
#define pb emplace_back
#define mp make_pair
#define all(x) (x).begin(), (x).end()

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    indexed_set;

const int N = 2e5 + 5;
const ll mod = 1e9 + 7;
const ll INf = 2e18;
const int P = 319;



int main()
{
    cin.sync_with_stdio(); cin.tie(0); cout.tie(0);
    ll n;
    cin >> n;
    cout << n - 1;

}
```

Task B ()

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

#define fi first
#define se second
#define sz(x) (int)(x).size()
#define pb push_back
#define mp make_pair
#define all(x) (x).begin(), (x).end()
//#define int long long

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    indexed_set;

const ll N = 3e5 + 5;
const ll mod = 1e9 + 7;
const ll INf = 2e18;
const int P = 319;

double Rast (double x1, double y1, double x2, double y2)
{
    return (x1 - x2) * (x1 - x2) + (y1 - y2) * (y1 - y2);
}

bool Ravn (double a, double b)
{
    return abs(a - b) <= 1;
}


void solve1()
{
    vector <double> a(6), b(6);
    for (int i = 0; i < 6; i++)
        cin >> a[i] >> b[i];
    vector <pair<double, int> > dist;
    for (int i = 1; i < 6; i++)
    {
        dist.pb({Rast(a[0], b[0], a[i], b[i]), i});
    }
    sort(all(dist));
    cout << setprecision(5) << fixed << a[0] << '␣' << b[0] << '␣' << a[dist[2].se] << '␣' << b[
        dist[2].se] << '␣' << a[dist[3].se] << '␣' << b[dist[3].se];

}

void solve2()
{
    vector <double> a(3), b(3), anx(6), any(6);
    for (int i = 0; i < 3; i++){
        cin >> a[i] >> b[i];
        anx[i * 2] = a[i];
        any[i * 2] = b[i];
    }
    for (int i = 0; i < 3; i++)
    {
        vector <double> x(2), y(2);
        int cnt = 0;
        for (int j = 0; j < 3; j++)
            if(i != j)
            {
                x[cnt] = a[j]; y[cnt] = b[j]; cnt++;
            }
        double X = (x[0] + x[1]) / 2, Y = (y[0] + y[1]) / 2;
        double vx = X - a[i], vy = Y - b[i];
        vx *= 4; vx /= 3; vy *= 4; vy /= 3;
        anx[(i * 2 + 3) % 6] = a[i] + vx;
        any[(i * 2 + 3) % 6] = b[i] + vy;
        //cout << i << ' ' << (i + 3) % 6 << '\n';
    }
    for (int i = 0; i < 6; i++)
        cout << setprecision(5) << fixed << anx[i] << '␣' << any[i] << '\n';
}

signed main()
{
    cin.sync_with_stdio(); cin.tie(0); cout.tie(0);

    /// /////////////////////////////
    //freopen ("qwwqqwwq", "r", stdin);
    /// /////////////////////////////

    int n;
    cin >> n;
    if(n == 6)
        solve1();
    else solve2();

}
```

Task C ()

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

#define fi first
#define se second
#define sz(x) (int)(x).size()
#define pb emplace_back
#define mp make_pair
#define all(x) (x).begin(), (x).end()

using namespace std;
using namespace __gnu_pbds;

typedef long long ll;
typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    indexed_set;

const int N = 1e4 + 5;
const ll mod = 1e9 + 7;
const ll INf = 2e18;
const int P = 319;

int bpos[N][26], n, l, was[N][505];
string t, s;

/*int dp (int pos, int kol_t)
{
    if(kol_t == l)
        return 0;
    if(pos == n)
    {
        return l - kol_t;
    }
    if(was[pos][kol_t] != -1)
        return was[pos][kol_t];
    int s_t = t[kol_t] - 'a';
    int an1 = dp(pos, kol_t + 1) + 1, an2 = mod;
    if(bpos[pos][s_t] != -1)
        an2 = dp(bpos[pos][s_t] + 1, kol_t + 1);
    was[pos][kol_t] = min(an1, an2);
    return min(an1, an2);
}*/


int main()
{
    cin.sync_with_stdio(); cin.tie(0); cout.tie(0);
    /// ////////////////////////////
    //freopen ("ikiki", "r", stdin);
    /// ////////////////////////////
    cin >> t;
    l = sz(t);
    int q;
    cin >> q;
    ll an = 0;
    for (int i = 0; i < N; i++)
        for (int j = 0; j < l + 3; j++)
            was[i][j] = -1;
    for (int ii = 0; ii < q; ii++)
    {
        cin >> s;
        n = sz(s);

        int ans = l;
        for (int i = 0; i < n; i++)
        {
            int kol = 0, pos = i, cnt = 0;
            while(true)
            {
                if(kol == l)
                    break;
                if(pos == n)
                {
                    cnt += l - kol;
                    break;
                }
                if(s[pos] == t[kol])
                {
                    pos++;
                    kol++;
                    continue;
                }
                kol++;
                cnt++;
                continue;
            }
            ans = min(ans, cnt);
        }
        an += ans;


    }
    cout << an;

}
```

Task D ()

Task E ()

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

#define fi first
#define se second
#define sz(x) (int)(x).size()
#define pb emplace_back
#define mp make_pair
#define all(x) (x).begin(), (x).end()
#define int long long

using namespace std;
//using namespace __gnu_pbds;

typedef long long ll;
//typedef tree<int, null_type, less<int>, rb_tree_tag, tree_order_statistics_node_update>
    indexed_set;

const ll N = 3e5 + 5;
const ll mod = 1e9 + 7;
const ll INf = 2e18;
const int P = 319;



signed main()
{
    cin.sync_with_stdio(); cin.tie(0); cout.tie(0);
    /// ////////////////////////////
    //freopen ("ikiki", "r", stdin);
    /// ////////////////////////////

    int n, m, b;
    cin >> n >> m >> b;
    vector <int> p(b), q(b);
    for (int i = 0; i < b; i++)
        cin >> p[i] >> q[i];
    vector <int> holst(1 << b);

    int cnt = 0;

    set <int> norm;
    for (int i = 0; i < (1 << b); i++)
        norm.insert(i);

    for (int i = 1; i <= b; i++)
    {
        set <int> qq = norm;
        for (int j = 0; j < (1 << (b - i)); j++)
        {
            int fir = *qq.begin();
            qq.erase(fir);
            int sec = *qq.begin();
            qq.erase(sec);
            ll q1 = N * (ll)fir + p[i - 1], q2 = N * (ll)sec + p[i - 1];
            cout << "?␣" << q1<< '␣' << q[i - 1] << '␣' << q2 << '␣' << q[i - 1] << endl;

            cnt++;
            assert(cnt <= 8191);
            int x, y;
            cin >> x >> y;
            int w = x / N;
            norm.erase(w);
        }
    }
    int pos = *norm.begin();
    cout << "!␣" << pos * N + 1 << '␣' << 1 << endl;
}
```

Task F ()