

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	100	25	525

Task A ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945 00:43:00
 */
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

// #define int long long

#pragma GCC optimize("Ofast")
#pragma GCC optimize("unroll-loops")
#pragma GCC target("sse, sse2, sse3, ssse3, sse4, popcnt, abm, mmx, avx, avx2, fma")

using namespace std;
using namespace __gnu_cxx;
using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
```

```

const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

void solve()
{
    //  ll n = sqrt((1LL << 20) - 1);
    //  cout << n << '\n';
    //  string an;
    //  while(n)
    //  {
    //      an += (n % 2 + '0');
    //      n /= 2;
    //  }
    //  reverse(all(an));
    //  cout << an;
    //  for (int n = 1; n < 26; n++) {
    //      if(n == 10) cout << "10 0\n";
    //      else cout << n << " : " << (n % 10 + (n / 10) % 10) % 10 << '\n';
    //  }
    int n;
    cin >> n;
    if(n < 10) {
        cout << n;
        re;
    }
    n -= 10;
    if(n % 9 == 0)
        cout << 0;
    else cout << n % 9 + 1;
}

signed main()
{
    //  freopen("input.txt", "r", stdin);
    //  freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int qq = 1;
    //  cin >> qq;
    rep(i, qq)
        solve();
}

```

Task B ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945  00:43:00
 */
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>
// #include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

// #define int long long

// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("sse, sse2, sse3, ssse3, sse4, popcnt, abm, mmx, avx, avx2, fma")

using namespace std;
// using namespace __gnu_cxx;
// using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

// vi tree;
//
// void Upd (int v, int L, int R, int pos, int k)
// {
//     if (R - L == 1)
//     {
//         tree[v] = k;
//         re;
//     }
//     int c = (L + R) / 2;
//     if (pos < c)
//         Upd(v * 2, L, c, pos, k);
```

```

//     else
//         Upd(v * 2 + 1, c, R, pos, k);
//     tree[v] = min(tree[v * 2], tree[v * 2 + 1]);
// }
//
// int Get (int v, int L, int R, int l, int r)
// {
//     if(L == l && R == r)
//         re tree[v];
//     int c = (L + R) / 2, an = mod;
//     if(l < c)
//         an = Get(v * 2, L, c, l, min(r, c));
//     if(c < r)
//         an = min(an, Get(v * 2 + 1, c, R, max(l, c), r));
//     re an;
// }

void solve()
{
    int n, m;
    cin >> n >> m;
    string s;
    cin >> s;
    //     tree.resize(n * 4);
    //     rep(i, n * 4)
    //         tree[i] = mod;
    vvi cnt(26);
    rep(i, 26)
    {
        cnt[i].resize(n);
        rep(j, n)
        {
            if(s[j] - 'a' == i) cnt[i][j] = 1;
            if(j) cnt[i][j] += cnt[i][j - 1];
        }
    }

    vi dp(n, mod);
    dp[0] = 1;
    //     Upd(1, 0, n, 0, 1);

    for (int i = 1; i < n; i++)
    {
        int L = max(-1, i - m), R = i;
        while(R - L > 1)
        {
            int M = (L + R) / 2;
            int kol = 0;
            rep(j, 26)
            {
                int now = cnt[j][i];
                if(M) now -= cnt[j][M - 1];
                if(now) kol++;
                if(kol > 3) break;
            }
            if(kol > 3)
                L = M;
            else R = M;
        }
        if(R == 0) dp[i] = 1;
        else
            dp[i] = 1 + dp[R - 1];
    }
    //     Upd(1, 0, n, i, dp[i]);
    cout << dp[n - 1];
}

signed main()
{
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int qq = 1;
    //     cin >> qq;

```

```
    rep(i, qq)
        solve();
}
```

Task C ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945  00:43:00
 */
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>
// #include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

// #define int long long

// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,fma")

using namespace std;
// using namespace __gnu_cxx;
// using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

// vi tree;
//
// void Upd (int v, int L, int R, int pos, int k)
// {
//     if (R - L == 1)
//     {
//         tree[v] = k;
//         re;
//     }
//     int c = (L + R) / 2;
//     if (pos < c)
//         Upd(v * 2, L, c, pos, k);
```

```

//     else
//         Upd(v * 2 + 1, c, R, pos, k);
//     tree[v] = min(tree[v * 2], tree[v * 2 + 1]);
// }
//
// int Get (int v, int L, int R, int l, int r)
// {
//     if(L == l && R == r)
//         re tree[v];
//     int c = (L + R) / 2, an = mod;
//     if(l < c)
//         an = Get(v * 2, L, c, l, min(r, c));
//     if(c < r)
//         an = min(an, Get(v * 2 + 1, c, R, max(l, c), r));
//     re an;
// }

int n, x, y;

int dp[2][250001];
int pr[501][250001];

void solve()
{
    cin >> n >> x >> y;
    vii a(n);
    rep(i, n)
        cin >> a[i].fi;
    rep(i, n)
        cin >> a[i].se;
    memset(dp, -1, sizeof(dp));
    int sumw = 0;
    rep(i, n)
        sumw += a[i].se;
    int need = sumw - y;

    dp[1][0] = 0;
    rep(i, n)
    {
        int now = i % 2;
        rep(v, x + 1)
        {
            if(dp[now ^ 1][v] != -1)
            {
                if(dp[now][v] < dp[now ^ 1][v])
                {
                    dp[now][v] = dp[now ^ 1][v];
                    pr[i][v] = v;
                }
                if(v + a[i].fi <= x && dp[now][v + a[i].fi] < dp[now ^ 1][v] + a[i].se)
                {
                    dp[now][v + a[i].fi] = dp[now ^ 1][v] + a[i].se;
                    pr[i][v + a[i].fi] = v;
                }
            }
            dp[now ^ 1][v] = -1;
        }
    }

    int an = -1, i, j;
    rep(v, x + 1)
    {
        // cout << dp[(n % 2) ^ 1][v] << '\n';
        if(an < dp[(n % 2) ^ 1][v])
        {
            an = dp[(n % 2) ^ 1][v];
            i = n - 1;
            j = v;
        }
    }
    // cout << an << '\n';
    if(an < need)
    {
        cout << -1;
    }
}

```

```

        re;
    }
    string ans;
    while(i >= 0)
    {
        int nj = pr[i][j];
        int ni = i - 1;
        if(j == nj)
            ans += 'y';
        else ans += 'x';
        i = ni;
        j = nj;
    }
    reverse(all(ans));
    cout << ans;

}

signed main()
{
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int qq = 1;
    //     cin >> qq;
    rep(i, qq)
        solve();
}

```


Task D ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945  00:43:00
 */
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>
// #include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

// #define int long long

// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,fma")

using namespace std;
// using namespace __gnu_cxx;
// using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

void solve()
{
    int n;
    cin >> n;
    n *= 2;
    string s;
    cin >> s;
    vi a(n);
    rep(i, n)
    {
        if(s[i] == '(' || s[i] == ')')
```

```

        a[i] = 1;
    }
    vi L(n), R(n);
    set<int> nice;
    rep(i, n)
    {
        L[i] = i - 1;
        R[i] = i + 1;
        if(i != n - 1 && a[i] == a[i + 1])
            nice.insert(i);
    }
    int cnt = n;
    while(sz(nice))
    {
        int el1 = *nice.begin();
        nice.erase(el1);
        int el2 = R[el1];
        nice.erase(el2);
        cnt -= 2;
        int Le = L[el1], Ri = R[el2];
        if(Le != -1 && Ri != n)
        {
            if(a[Le] == a[Ri])
                nice.insert(Le);
            else if(a[Le] == a[el1])
                nice.erase(Le);
            R[Le] = Ri;
            L[Ri] = Le;
        }
        else if(Le != -1)
        {
            if(a[Le] == a[el1])
                nice.erase(Le);
            R[Le] = n;
        }
        else if(Ri != n)
        {
            L[Ri] = -1;
        }
    }
    cout << cnt / 2;
}

signed main()
{
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int qq = 1;
    // cin >> qq;
    rep(i, qq)
        solve();
}

```

Task E ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945  00:43:00
 */
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>
// #include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

// #define int long long

// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,fma")

using namespace std;
// using namespace __gnu_cxx;
// using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

map<vi, int> an1, an2;

void solve2()
{
    set<vi> was;
    for (int i = 1; i <= 10; i++)
        for (int j = i + 1; j <= 10; j++)
            for (int h = j + 1; h <= 10; h++)
                {
                    bool f1 = 0;
                    for (int ne = 1; ne <= 10; ne++)
                        {
```

```

        if(ne == i || ne == j || ne == h) continue;
        vi now = {i, j, h, ne};
        sort(all(now));
        if(was.find(now) == was.end())
        {
            was.insert(now);
            an1[{i, j, h}] = ne;
            an2[now] = ne;
            fl = 1;
            break;
        }
    }
}

// for (auto x : an)
// cout << x.fi[0] << ' ' << x.fi[1] << ' ' << x.fi[2] << ' ' << x.fi[3] << " : " << x.se
// << '\n';
}

void solveadd2 (int n, int k, vi& a)
{
    sort(all(a));
    cout << an1[a] << '\n';
}

void solveclear2 (int n, int k, vi& a)
{
    sort(all(a));
    int el = an2[a];
    rep(i, k + 1) {
        if(a[i] != el)
            cout << a[i] << '␣';
    }
}

vi p;

void solve1()
{
    for (int i = 1; i <= 1000000; i++)
        p.pb(i);
    random_shuffle(all(p));
}

void solveadd1 (int n, int k, vi& a)
{
    solve1();
    cout << p[0] << '\n';
}

void solveclear1 (int n, int k, vi& a)
{
    solve1();
    rep(i, k + 1)
        if(a[i] != p[0])
            cout << a[i] << '␣';
    cout << '\n';
}

vi per, obr;

void solve3()
{
    per.resize(100000 + 1);
    obr.resize(100000 + 1);
    p.clear();
    rep(i, 100000)
        p.pb(i + 1);
    random_shuffle(all(p));
    rep(i, 100000)
    {
        per[i + 1] = p[i];
        obr[p[i]] = i + 1;
    }
}

```

```

vi now;
int nn, Ma;

void go (int pos)
{
    if(pos == nn)
    {
        for (int i = 1; i <= Ma; i++)
        {
            bool nice = 1;
            rep(j, nn)
            {
                if(i == now[j])
                {
                    nice = 0;
                    break;
                }
            }
            if(nice)
            {
                vi now1 = now;
                now1.pb(i);
                sort(all(now1));
                if(an2.find(now1) == an2.end())
                {
                    an1[now] = i;
                    an2[now1] = i;
                    re;
                }
            }
        }
        cout << ":";
        exit(0);
        re;
    }
    int el;
    if(pos)
        el = now.back();
    else
        el = 0;
    for (int ne = el + 1; ne <= Ma - (nn - pos) + 1; ne++)
    {
        now.pb(ne);
        go(pos + 1);
        now.pop_back();
    }
}

void solveadd3 (int n, int k, vi& a)
{
    rep(i, k) {
        a[i] = per[a[i]];
    }
    vi was(n + 1);
    rep(i, k)
        was[a[i]] = 1;
    vi Now;
    for (int i = 1; i <= 15; i++)
        if(was[i])
            Now.pb(i);
    nn = sz(Now);
    // assert(nn < 8);
    Ma = 15;
    an1.clear();
    an2.clear();
    go(0);
    cout << obr[an1[Now]] << '\n';
}

void solveclear3 (int n, int k, vi& a)
{
    rep(i, k + 1) {
        a[i] = per[a[i]];
    }
}

```

```

    }
    vi was(n + 1);
    rep(i, k + 1)
        was[a[i]] = 1;
    vi Now;
    for (int i = 1; i <= 15; i++)
        if(was[i])
            Now.pb(i);
    nn = sz(Now) - 1;
    Ma = 15;
    an1.clear();
    an2.clear();
    go(0);
    int el = an2[Now];
    rep(i, k + 1)
    {
        if(a[i] != el)
            cout << obr[a[i]] << '␣';
    }
    cout << '\n';
}

void solveadd()
{
    int n, k;
    cin >> n >> k;
    vi a(k);
    rep(i, k)
        cin >> a[i];
    if(n == 10) {
        solve2();
        solveadd2(n, k, a);
    }
    if(n == 1000000) {
        solveadd1(n, k, a);
    }
    if(n == 100000) {
        if(!sz(per))
            solve3();
        solveadd3(n, k, a);
    }
}

void solveclear()
{
    int n, k;
    cin >> n >> k;
    vi a(k + 1);
    rep(i, k + 1)
        cin >> a[i];
    if(n == 10) {
        solve2();
        solveclear2(n, k, a);
    }
    if(n == 1000000) {
        solveclear1(n, k, a);
    }
    if(n == 100000) {
        if(!sz(per))
            solve3();
        solveclear3(n, k, a);
    }
}

signed main()
{
    // freopen("input.txt", "r", stdin);
    // freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    srand(1654444);
    int qq = 1;
    string s;
    cin >> s;
    cin >> qq;

```

```
rep(i, qq) {  
    if(s == "add")  
        solveadd();  
    else  
        solveclear();  
}
```

Task F ()

```
/**
 *   author:   tourist
 *   created:  09.05.1945  00:43:00
 */
#include <bits/stdc++.h>
// #include <ext/pb_ds/assoc_container.hpp>
// #include <ext/pb_ds/tree_policy.hpp>

#define fi first
#define se second
#define pb emplace_back
#define all(x) x.begin(), x.end()
#define re return
#define mp make_pair
#define sz(x) ((int)(x).size())
#define rep(i, n) for (int i = 0; i < (n); i++)
#define rrep(i, n) for (int i = (n) - 1; i >= 0; i--)
#define sqr(x) ((x)*(x))

#define unq(x); { sort(x.begin(), x.end()); x.resize(unique(x.begin(), x.end()) - x.begin()); }

#define int long long

// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC target("sse, sse2, sse3, ssse3, sse4, popcnt, abm, mmx, avx, avx2, fma")

using namespace std;
// using namespace __gnu_cxx;
// using namespace __gnu_pbds;

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

typedef long long ll;
typedef vector<int> vi;
typedef vector<vi> vvi;
typedef pair<int, int> ii;
typedef vector<ii> vii;
typedef vector<string> vs;
typedef pair<ll, ll> pll;
typedef vector<ll> vll;
typedef complex<double> cmp;
typedef vector<bool> vb;
typedef vector<vector<char>> vvc;
typedef double D;

double GetTime() { return clock() / (double) CLOCKS_PER_SEC; };
mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());

const int mod = 1e9 + 7;
const ll INF = 3e18;
const int N = 1e6 + 5;
const int base0 = 437;
const int base1 = 343;
const int d = 400;
const D pi = acos(-1);
const long double eps = 1e-4;

void solve8(int n)
{
    cout << "4\n0_0\n0_1\n1_1\n1_0\n";
    int dx[8] = {1, 1, 1, 0, 0, -1, -1, -1};
    int dy[8] = {-1, 0, 1, -1, 1, -1, 0, 1};
    rep(i, n)
    {
        cout << dx[i] << '_' << dy[i] << '\n';
    }
}

void solve()
```



```

{
    int n;
    cin >> n;
    solve8(n);
}

signed main()
{
    //     freopen("input.txt", "r", stdin);
    //     freopen("output.txt", "w", stdout);
    ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
    int qq = 1;
    //     cin >> qq;
    rep(i, qq)
        solve();
}

```