

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	35	25	460

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef long long ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;

int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif
    ll k;
    cin >> k;
    ll prev = 1;
    for (ll i = 2; i <= k; i++) {
        prev = i + (prev / 10);
    }
    cout << prev % 10;
}
```

Task B ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef long long ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;

set <char> ss;
int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif
    ll k, n;
    cin >> n >> k;
    string s;
    cin >> s;
    ll ans = 0;
    for (ll i = 0; i < n; i++) {
        ll kolvo = 0;
        ss.clear();
        while (i < n) {
            ss.insert(s[i]);
            kolvo++;
            if (kolvo > k || ss.size() > 3) {
                i--;
                break;
            }
            i++;
        }
        ans++;
    }
    cout << ans;
}
```

Task C ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef int ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;

ll dp[501][250001];
bool p[501][250001];
bool pl[501][250001];
pair <ll, ll> A[N];

void f(ll n, ll x) {
    pl[n][x] = 1;
    if (n == 0) {
        dp[n][x] = 0;
        return;
    }
    if (!pl[n - 1][x]) {
        f(n - 1, x);
    }
    dp[n][x] = dp[n - 1][x] + A[n - 1].second;
    if (A[n - 1].first > x) {
        return;
    }
    if (!pl[n - 1][x - A[n - 1].first]) {
        f(n - 1, x - A[n - 1].first);
    }
    if (dp[n][x] > dp[n - 1][x - A[n - 1].first]) {
        dp[n][x] = dp[n - 1][x - A[n - 1].first];
        p[n][x] = 1;
    }
}

int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif

    ll n, x, y;
    cin >> n >> x >> y;
    for (ll i = 0; i < n; i++) {
        cin >> A[i].first;
    }
    for (ll i = 0; i < n; i++) {
        cin >> A[i].second;
    }
    f(n, x);
    if (dp[n][x] > y) {
        cout << -1;
        return 0;
    }
    string ans;
    ll curn = n, curw = x;
    for (ll i = n; i >= 1; i--) {
        if (p[i][curw]) {
            ans.push_back('x');
            curw -= A[i - 1].first;
        }
    }
```

```
        else {
            ans.push_back('y');
        }
    }
    reverse(ans.begin(), ans.end());
    for (ll i = 0; i < n; i++) {
        cout << ans[i];
    }
}
```

Task D ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef int ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;

ll A[N];
string s;

int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif
    ll n;
    cin >> n;
    cin >> s;
    for (ll i = 0; i < 2 * n; i++) {
        if (s[i] == '(' || s[i] == ')') {
            A[i] = 1;
        }
        else {
            A[i] = 2;
        }
    }
    stack<ll> ss;
    ll ans = 0;
    for (ll i = 0; i < 2 * n; i++) {
        if (ss.empty()) {
            ss.push(A[i]);
            continue;
        }
        if (ss.top() == A[i]) {
            ss.pop();
            continue;
        }
        if (ss.size() < 2 * n - i) {
            ss.push(A[i]);
            continue;
        }
        ans++;
        ss.pop();
    }
    cout << ans;
}
```

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef int ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;

ll A[N], V1[N], V2[N], C[N], D[N];
int n, k;
vector<vector<int>>> g;
vector<int> mt;
vector<char> used;

bool try_kuhn(int v) {
    if (used[v]) return false;
    used[v] = true;
    for (size_t i = 0; i < g[v].size(); ++i) {
        int to = g[v][i];
        if (mt[to] == -1 || try_kuhn(mt[to])) {
            mt[to] = v;
            return true;
        }
    }
    return false;
}

int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif

    ll cnt = 0, cnt1 = 0;
    for (ll i = 0; i < 1e5; i++) {
        A[i] = -1;
    }
    vector<ll> v1;
    g.resize(331);
    for (ll i = 0; i < 10; i++) {
        for (ll j = i + 1; j < 10; j++) {
            for (ll q = j + 1; q < 10; q++) {
                ll x = i + 10 * j + 100 * q;
                V1[cnt] = x;
                A[x] = cnt++;
                v1.push_back(x);
                for (ll r = 0; r < 10; r++) {
                    if (r == i || r == j || r == q)
                        continue;
                    ll B[4];
                    B[0] = i; B[1] = j; B[2] = q; B[3] = r;
                    sort(B, B + 4);
                    ll y = B[0] + B[1] * 10 + B[2] * 100 + B[3] * 1000;
                    if (A[y] == -1) {
                        V2[cnt1] = y;
                        A[y] = cnt1++;
                        v1.push_back(y);
                    }
                }
                g[A[x]].push_back(A[y]);
            }
        }
    }
}
```

```

    }
}

}
n = 120, k = 210;
mt.assign(k, -1);
for (int v = 0; v < n; ++v) {
    used.assign(n, false);
    try_kuhn(v);
}
vector<ll> mt1(120);
for (ll i = 0; i < 210; i++) {
    if (mt[i] != -1) {
        mt1[mt[i]] = i;
    }
}
string s;
cin >> s;
if (s == "add") {
    ll t;
    cin >> t;
    while (t--) {
        ll n1, k1;
        cin >> n1 >> k1;
        if (n1 == 10) {

            for (ll i = 3; i > 0; i--) {
                cin >> C[i - 1];
                C[i - 1]--;
            }
            sort(C, C + 3);
            ll a = C[0] + 10 * C[1] + 100 * C[2];
            ll b = V2[mt1[A[a]]];
            for (ll i = 0; i < 4; i++) {
                if (b % 10 == C[0] || b % 10 == C[1] || b % 10 == C[2]) {
                    b /= 10;
                    continue;
                }
                cout << b % 10 + 1 << '\n';
                break;
            }
            cout.flush();
        }
        if (n1 == 1e6) {
            for (ll i = 0; i < k1; i++) {
                cin >> C[i];

                C[i]--;
            }
            sort(C, C + k1);
            ll cur = 0;
            bool w = 0;
            for (ll i = 0; i < 1e5; i++) {
                ll kolvo = 0;
                ll nach = cur;
                while (cur < k1) {
                    if (C[cur] >= 10 * i + 10) {
                        break;
                    }
                    C[cur] %= 10;
                    kolvo++;
                    cur++;
                }
                if (kolvo == 3) {
                    sort(C + nach, C + nach + 3);
                    ll a = C[nach] + 10 * C[nach + 1] + 100 * C[nach + 2];
                    ll b = V2[mt1[A[a]]];
                    for (ll j = 0; j < 4; j++) {
                        if (b % 10 == C[nach] || b % 10 == C[nach + 1] || b % 10 == C[nach + 2]) {
                            b /= 10;
                            continue;
                        }
                    }
                }
            }
        }
    }
}

```

```

        cout << b % 10 + 1 + 10 * i << '\n';
        break;
    }
    cout.flush();
    w = 1;
    break;
}
}
if (!w) {
    return 1;
}
}
}
else {
    ll t;
    cin >> t;
    while (t--) {
        ll n1, k1;
        cin >> n1 >> k1;
        if (n1 == 10) {
            for (ll i = 4; i > 0; i--) {
                cin >> C[i - 1];
                C[i - 1]--;
            }
            sort(C, C + 4);
            ll a = C[0] + 10 * C[1] + 100 * C[2] + 1000 * C[3];
            ll b = V1[mt[A[a]]];
            for (ll i = 0; i < 3; i++) {
                cout << b % 10 + 1 << '\n';
                b /= 10;
            }
            cout.flush();
        }
        if (n1 == 1e6) {
            for (ll i = 0; i < k1 + 1; i++) {
                cin >> C[i];
                C[i]--;
            }
            sort(C, C + k1);
            for (ll i = 0; i < k1 + 1; i++) {
                D[i] = C[i];
            }
            ll cur = 0;
            bool w = 0;
            for (ll i = 0; i < 1e5; i++) {
                ll kolvo = 0;
                ll nach = cur;
                while (cur < k1 + 1) {
                    if (C[cur] >= 10 * i + 10) {
                        break;
                    }
                    C[cur] %= 10;
                    kolvo++;
                    cur++;
                }
                if (kolvo == 4) {
                    sort(C + nach, C + nach + 4);
                    ll a = C[nach] + 10 * C[nach + 1] + 100 * C[nach + 2] + 1000 * C[nach + 3];
                    ll b = V1[mt[A[a]]];
                    ll b1 = b;
                    bool p = 0;
                    for (ll j = 0; j < 3; j++) {
                        if (b % 10 != C[nach] && b % 10 != C[nach + 1] && b % 10 != C[nach + 2] && b % 10 != C[nach + 3]) {
                            p = 1;
                            break;
                        }
                    }
                    b /= 10;
                }
                if (p)
                    continue;
            }
        }
    }
}

```


Task F ()

```
#define _CRT_SECURE_NO_WARNINGS

#include <iostream>
#include <vector>
#include <algorithm>
#include <set>
#include <math.h>
#include <string>
#include <iomanip>
#include <queue>
#include <stack>
#include <map>

using namespace std;

typedef long long ll;
typedef long double ld;

const ll N = 2e5 + 1;
const ll inf = 1e18;
const ld pi = acos(-1);

int main() {
#ifdef _DEBUG
    freopen("input.txt", "r", stdin);
    //freopen("output.txt", "w", stdout);
#endif

    ll n;
    cin >> n;
    if (n == 1) {
        cout << 2 * 2 << '\n';
        for (ll i = 0; i < 2 * 2; i++) {
            ld k = ld(i) / ld(2) * pi;
            cout << fixed << ll(cos(k) * 1e15) << " " << ll(sin(k) * 1e15) << '\n';
        }
        for (ll i = 0; i < 2 * 1; i += 2) {
            ld k = ld(i) / ld(2) * pi;
            cout << ll(2 * cos(k) * 1e15) << " " << ll(2 * sin(k) * 1e15) << '\n';
        }
        return 0;
    }
    cout << 4 << '\n';
    cout << 1 << " " << 1 << '\n';
    cout << -1 << " " << 1 << '\n';
    cout << -1 << " " << -1 << '\n';
    cout << 1 << " " << -1 << '\n';
    pair<ll, ll> p[] = { {2, 0}, {-2, 0}, {0, -2}, {0, 2}, {2, 2}, {-2, 2}, {-2, -2}, {2, -2} };
    for (ll i = 0; i < min(ll(8), n); i++) {
        cout << p[i].first << " " << p[i].second << '\n';
    }
}
```