

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	75	100	16	10	401

Task A ()

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

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;
const int sz = 10;

int d[sz];
bool fl = 0;
int ans[sz];
int crt[sz];
bool used[sz];
int nd[sz];

bool check() {
    for (int i = 0; i < 6; ++i) {
        nd[i] = 1;
        for (int j = 0; j < i; ++j) {
            if (crt[j] < crt[i]) {
                ++nd[i];
            }
        }
        if (d[i] != nd[i]) return 0;
    }
    return 1;
}

void f(int t) {
    if (t == 6) {
        if (check()) {
            fl = 1;
            for (int i = 0; i < 6; ++i) {
                ans[i] = crt[i];
            }
        }
        return;
    }
    for (int i = 1; i <= 6 && !fl; ++i) {
        if (!used[i]) {
            used[i] = 1;
            crt[t] = i;
            f(t + 1);
            used[i] = 0;
        }
    }
}

int main() {
    for (int i = 0; i < 6; ++i) {
        cin >> d[i];
    }
    f(0);
    for (int i = 0; i < 6; ++i) {
        cout << ans[i] << ' ';
    }
}
```

} }

Task B ()

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

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;
const int sz = 100100;

ll a[sz];

void first() {
    int n;
    cin >> n;
    ll s = 0;
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
        s += a[i];
    }
    cout << s * 1000 << endl;
}

void second() {
    int n;
    cin >> n;
    ll s = 0;
    for (int i = 0; i < n; ++i) {
        cin >> a[i];
        s += a[i] % 1000;
    }
    cout << s + a[0] / 1000 << endl;
}

int main() {
    string s;
    cin >> s;
    if (s == "first") {
        first();
    } else {
        second();
    }
}
```

Task C ()

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

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;
const int sz = 10;

set <pair <int, int> > s;
int a[sz], b[sz];

void add(int a, int b) {
    if (a <= 0 || b <= 0) return;
    if (a > b) swap(a, b);
    s.insert({a, b});
}

void two(int i, int j) {
    if (a[i] < a[j] || b[i] < b[j]) return;
    if (a[i] == a[j] && b[i] == b[j]) return;
    if (a[i] == a[j]) add(a[i], b[i] - b[j]);
    if (b[i] == b[j]) add(a[i] - a[j], b[i]);
}

void three(int i, int j, int k) {
    if (a[i] < a[j] || b[i] < b[j]) return;
    if (a[i] == a[j] && b[i] == b[j]) return;
    if (a[i] < a[k] || b[i] < b[k]) return;
    if (a[i] == a[k] && b[i] == b[k]) return;
    if (a[i] == a[j]) {
        if (b[i] == b[j] + b[k]) add(a[i] - a[k], b[k]);
        if (a[i] == a[k]) add(a[i], b[i] - b[j] - b[k]);
        return;
    }
    if (b[i] == b[j]) {
        if (a[i] == a[k] + a[j]) add(a[k], b[i] - b[k]);
        if (b[i] == b[k]) add(b[i], a[i] - a[j] - a[k]);
        return;
    }
    if (a[j] == a[k] && b[j] + b[k] == b[i]) {
        add(a[i] - a[j], b[i]);
    }
    if (b[j] == b[k] && a[k] + a[j] == a[i]) {
        add(a[i], b[i] - b[j]);
    }
    if (a[j] + a[k] == a[i] && b[i] <= b[j] + b[k]) {
        add(a[k], b[i] - b[k]);
        add(a[j], b[i] - b[j]);
    }
    if (b[j] + b[k] == b[i] && a[i] <= a[j] + a[k]) {
        add(b[j], a[i] - a[j]);
        add(b[k], a[i] - a[k]);
    }
    if (a[i] > a[j] + a[k] && b[i] < b[j] + b[k]) {
        add(a[j], b[i] - b[j]);
    }
    if (b[i] > b[j] + b[k] && a[i] < a[j] + a[k]) {
        add(b[j], a[i] - a[j]);
    }
    if (a[k] == a[i]) {
        add(a[i], b[i] - b[j] - b[k]);
        add(b[j], a[i] - a[j]);
    }
    if (b[k] == b[i]) {
        add(a[i] - a[k] - a[j], b[i]);
        add(a[j], b[i] - b[j]);
    }
}

void rev(int i) {
    swap(a[i], b[i]);
}
```

```

}

int main() {
    for (int i = 0; i < 3; ++i) {
        cin >> a[i] >> b[i];
        add(a[i], b[i]);
    }
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            if (i == j) continue;
            two(i, j);
            rev(j);
            two(i, j);
            rev(i);
            two(i, j);
            rev(j);
            two(i, j);
        }
    }
    for (int i = 0; i < 3; ++i) {
        for (int j = 0; j < 3; ++j) {
            if (i == j) continue;
            for (int k = 0; k < 3; ++k) {
                if (k == i || k == j) continue;
                three(i, j, k);
                rev(k);
                three(i, j, k);
            }
            rev(j);
            for (int k = 0; k < 3; ++k) {
                if (k == i || k == j) continue;
                three(i, j, k);
                rev(k);
                three(i, j, k);
            }
        }
        rev(i);
        for (int j = 0; j < 3; ++j) {
            if (i == j) continue;
            for (int k = 0; k < 3; ++k) {
                if (k == i || k == j) continue;
                three(i, j, k);
                rev(k);
                three(i, j, k);
            }
            rev(j);
            for (int k = 0; k < 3; ++k) {
                if (k == i || k == j) continue;
                three(i, j, k);
                rev(k);
                three(i, j, k);
            }
        }
    }
    for (auto p : s) {
        cout << p.first << ' ' << p.second << '\n';
    }
}

```

Task D ()

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

using namespace std;

typedef long long ll;

const ll mod = 1e9 + 7;
const int sz = 110;

int d[sz][sz];
int a[sz];
int crt[sz];

int main() {
    int n;
    cin >> n;
    for (int i = 1; i <= n; ++i) {
        cin >> a[i];
        crt[i] = 2 * a[i] + 1;
    }
    for (int i = 1; i <= n; ++i) {
        for (int j = 0; j <= a[i]; ++j) {
            d[i][j] = j;
        }
        for (int j = a[i] + 1; j <= a[i] * 2 + 1; ++j) {
            vector<int> v(1, a[i]);
            for (int k = a[i] + 1; k < j; ++k) {
                v.push_back(d[i][k]);
            }
            v.push_back(1000);
            sort(v.begin(), v.end());
            if (v[0] != 0) {
                d[i][j] = 0;
            } else {
                d[i][j] = -1;
                for (int k = 0; k < v.size() - 1 && d[i][j] == -1; ++k) {
                    if (v[k + 1] - v[k] > 1) {
                        d[i][j] = v[k] + 1;
                    }
                }
            }
        }
    }
    int x = 0;
    for (int i = 1; i <= n; ++i) {
        x ^= d[i][a[i] * 2 + 1];
    }
    if (x == 0) {
        cout << "-1_-1" << endl;
        return 0;
    }
    int pos = 0, cnt = 0;
    while (true) {
        bool fl = 0;
        for (int i = 1; i <= n && !fl; ++i) {
            if (crt[i] == a[i] + 1 || crt[i] == 0) continue;
            if (crt[i] > a[i]) {
                for (int j = a[i] + 1; j < crt[i] && !fl; ++j) {
                    if ((x ^ d[i][crt[i]] ^ d[i][j]) == 0) {
                        fl = 1;
                        cout << i << ' ' << crt[i] - j << endl;
                        crt[i] = j;
                    }
                }
            }
        }
        if (fl) continue;
        for (int j = 0; j < crt[1] && !fl; ++j) {
            if ((x ^ d[1][crt[1]] ^ d[1][j]) == 0) {
                fl = 1;
                cout << 1 << ' ' << crt[1] - j << endl;
                crt[1] = j;
            }
        }
    }
}
```

```

    }
}
for (int i = 1; i <= n && !fl; ++i) {
    if (crt[i] <= a[i]) continue;
    if ((x ^ a[i] ^ d[i][crt[i]]) == 0) {
        crt[i] = a[i];
        cout << i << ' ' << 0 << endl;
        fl = 1;
    }
}
cin >> pos >> cnt;
if (pos == -1) return 0;
x = d[pos][crt[pos]];
if (cnt == 0) {
    crt[pos] = a[pos];
} else {
    crt[pos] -= cnt;
}
x ^= d[pos][crt[pos]];
}
}

```

Task E ()

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

using namespace std;

typedef long long ll;

string s[10];

void clean() {
    for (int i = 0; i < 10; ++i) {
        for (int j = 0; j < 10; ++j) {
            s[i][j] = '0';
        }
    }
}

void print() {
    for (int i = 0; i < 10; ++i) {
        cout << s[i] << '\n';
    }
    cout << endl;
}

void first() {
    int n;
    cin >> n;
    clean();
    if (n == 1024) {
        print();
        return;
    }
    int pos, cnt;
    bool fl = (n > 1024);
    for (int i = 0; i < 10; ++i) {
        if (n % 2) {
            for (int j = 0; j <= i; ++j) {
                s[i][j] = '1';
            }
            cnt = i;
        } else {
            pos = i;
        }
        n /= 2;
    }
    if (fl) {
        for (int j = 0; j <= cnt; ++j) {
            s[pos][j] = '1';
        }
    }
    print();
}

void second() {
    ll n = 0;
    set<int> st;
    for (int i = 0; i < 10; ++i) {
        cin >> s[i];
        int cnt = 0;
        for (int j = 0; j < 10; ++j) {
            if (s[i][j] == '1') ++cnt;
        }
        if (cnt != 0) {
            if (st.find(cnt) != st.end()) cnt = 11;
            n += (1 << (cnt - 1));
            st.insert(cnt);
        }
    }
    if (n == 0) n = 1024;
    cout << n << endl;
}

int main() {
```



```

    for (int i = 0; i < 10; ++i) {
        for (int j = 0; j < 10; ++j) {
            s[i] += '0';
        }
    }
    int t;
    cin >> t;
    string s;
    cin >> s;
    if (s == "transmit") {
        while (t--) {
            first();
            cout << endl;
        }
    } else {
        while (t--) {
            second();
        }
    }
}

```

Task F ()

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

using namespace std;

typedef long long ll;
const int sz = 300300;

struct otr {
    otr(ll len_, int c_): len(len_), c(c_) {}
    ll len;
    int c;
};

otr get_otr(string& a, int it) {
    string num = "";
    --it;
    while (a[it] != '|' ) {
        num += a[it];
        --it;
    }
    --it;
    ll len = 0;
    for (int i = num.size() - 1; i >= 0; --i) {
        len = len * 10 + (num[i] - '0');
    }
    return otr(len, a[it] - '0');
}

void build(string& a, vector<otr>& x) {
    for (int i = a.size() - 1; i >= 0; --i) {
        if (a[i] != ')') {
            x.push_back(otr(1, a[i] - '0'));
        } else {
            x.push_back(get_otr(a, i));
            while (a[i] != '(') --i;
        }
    }
}

void rebuild(vector<otr>& x, vector<ll>& gr) {
    vector<otr> tmp;
    ll tl = 0;
    int it = 1;
    for (auto t : x) {
        tl += t.len;
        while (it < gr.size() && gr[it] <= tl) {
            tmp.push_back(otr(gr[it] - gr[it - 1], t.c));
            ++it;
        }
    }
    while (it < gr.size()) {
        tmp.push_back(otr(gr[it] - gr[it - 1], 0));
        ++it;
    }
    x = tmp;
}

int main() {
    string a, b;
    cin >> a >> b;
    vector<otr> x, y;
    build(a, x);
    build(b, y);
    set<ll> s;
    ll tl = 0;
    for (auto t : x) {
        tl += t.len;
        s.insert(tl);
    }
    tl = 0;
    for (auto t : y) {
        tl += t.len;
```

```

        s.insert(tl);
    }
    vector<ll> gr(1, 0);
    for (ll i : s) {
        gr.push_back(i);
    }
    // for (auto t : x) {
    //     cout << '(' << t.c << ' ' << '|' << t.len << " ";
    // }
    // cout << '\n';
    // for (auto t : y) {
    //     cout << '(' << t.c << ' ' << '|' << t.len << " ";
    // }
    // cout << '\n';
    rebuild(x, gr);
    rebuild(y, gr);
    // for (auto t : x) {
    //     cout << '(' << t.c << ' ' << '|' << t.len << " ";
    // }
    // cout << '\n';
    // for (auto t : y) {
    //     cout << '(' << t.c << ' ' << '|' << t.len << " ";
    // }
    // cout << '\n';
    vector<otr> ans;
    bool fl = 0, next = 0;
    for (int i = 0; i < x.size(); ++i) {
        fl = next;
        auto p = x[i], t = y[i];
        otr res(p.len, 0);
        next = 0;
        if (p.c + t.c < 10) {
            res.c = p.c + t.c;
        } else {
            next = 1;
            res.c = (p.c + t.c) % 10 + 1;
            --res.len;
            otr n(1, res.c - 1);
            if (fl) ++n.c;
            ans.push_back(n);
            ans.push_back(res);
            continue;
        }
        if (!fl) {
            ans.push_back(res);
            continue;
        }
        if (res.c == 9) {
            res.c = 0;
            next = 1;
            ans.push_back(res);
            continue;
        }
        otr n(1, res.c + 1);
        res.len--;
        ans.push_back(n);
        ans.push_back(res);
    }
    if (next) {
        ans.push_back({1, 1});
    }
    // for (auto t : ans) {
    //     cout << '(' << t.c << ' ' << '|' << t.len << " ";
    // }
    reverse(ans.begin(), ans.end());
    for (auto p : ans) {
        if (p.len <= 0) continue;
        cout << '(' << p.c << '|' << p.len << ')';
    }
}

```