

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	16	10	386

Task A ()

```
#include <bits/stdc++.h>
using namespace std;

signed main() {
    int n= 6;
    vector<int> a(n);
    for (int i = 0; i < n; i++) cin >> a[i];
    vector<int> g;
    g.push_back(0);
    vector<int> ans(6);
    ans[0] = 1;
    for (int i = 1; i < 6; i++) {
        auto pos = g.begin();
        for (int x = 1; x < a[i]; x++) pos++;
        for (auto j = pos; j < g.end(); j++) ans[*j]++;
        g.insert(pos, i);
        ans[i] = a[i];
    }
    for (int i = 0; i < n; i++) cout << ans[i] << ' ';
}
```

Task B ()

```
#include <bits/stdc++.h>
using namespace std;

signed main() {
    string pl; cin >> pl;
    if (pl == "first") {
        int n; cin >> n;
        int sum = 0;
        for (int j = 0; j < n; j++) {
            int x; cin >> x;
            sum += x;
        }
        cout << sum * 228;
    } else {
        int n; cin >> n;
        int result = 0;
        int xx = 0;
        for (int i = 0; i < n; i++) {
            int x; cin >> x;
            cout << x % 228 << " moded\n";
            result += (x % 228);
            xx = x / 228;
        }
        cout << xx << " xx\n";
        cout << result + xx;
    }
}
```

Task C ()

```
#include <bits/stdc++.h>
using namespace std;
vector <pair<int, int>> t;
set<pair<int, int>> ans;
void add(int x, int y) {
//    cout << " add " << x << ' ' << y << '\n';
    if (x <= 0 || y <= 0) return;
    if (x > y) swap(x, y);
    ans.insert({x, y});
//    cout << " ans inserted! " << ans.size() << '\n';
}
vector <vector <bool>> target;
vector <vector <bool>> used;
int minx, maxx, miny, maxy;
int squ;
vector <pair<int, int>> d = {{1, 0}, {0, 1}, {-1, 0}, {0, -1}};
void dfs(int x, int y) {
//    cout << " dfs " << x << ' ' << y << '\n';
    used[x][y] = 1;
    squ++;
    minx = min(minx, x);
    maxx = max(maxx, x);
    miny = min(y, miny);
    maxy = max(maxy, y);
    for (auto [dx, dy]: d) {
        if (x + dx >= 0 && x + dx < target.size() && y + dy >= 0 && y + dy < target[0].size()) {
            if (!used[x + dx][y + dy] && !target[x + dx][y + dy]) {
                dfs(x + dx, y + dy);
            }
        }
    }
}
void dfsix() {
//    cout << " dfsix\n";
//    for (auto i: target) {
//        for (int j: i) cout << j;
//        cout << '\n';
//    }
    used.assign(target.size(), vector <bool> (target[0].size(), 0));
    for (int i = 0; i < target.size(); i++) {
        for (int j = 0; j < target[0].size(); j++) {
            if (!used[i][j] && !target[i][j]) {
//                cout << " cooldown\n";
                squ = 0;
                minx = 1e9, maxx = 0;
                miny = 1e9, maxy = 0;
                dfs(i, j);
                if (squ == (maxx - minx + 1) * (maxy - miny + 1)) {
                    add(maxx - minx + 1, maxy - miny + 1);
                }
            }
        }
    }
}
void check2() {
//    cout << " called check2\n";
    if (t[0].first == t[1].first) {
        add(t[0].first, t[0].second - t[1].second);
    }
    if (t[0].second == t[1].second) {
        add(t[0].first - t[1].first, t[1].second);
    }
//    for (auto [x, y]: t) cout << x << ' ' << y << '\n';
    target.assign(t[0].first, vector <bool> (t[0].second, 0));
    for (int i = 0; i < t[1].first; i++) {
        for (int j = 0; j < t[1].second; j++) {
            target[i][j] = 1;
        }
    }
    if (t[1].first == t[2].first && t[1].second >= t[2].second) {
        add(t[2].first, t[2].second);
    }
}
```

```

        add(t[2].first , t[1].second - t[2].second);
    }
    if (t[1].second == t[2].second && t[1].first >= t[2].first) {
        add(t[2].first , t[2].second);
        add(t[2].second , t[1].first - t[2].first);
    }
    for (int x = 0; x < t[0].first; x++) {
        for (int y = 0; y < t[0].second; y++) {
//             cout << x << ' ' << y << '\n';
            if (x < t[1].first && y < t[1].second) continue;
            if (x + t[2].first > t[0].first || y + t[2].second > t[0].second) continue;
//             cout << "alive\n";
            bool ar = (x == t[1].first || x == 0 || x + t[2].first == t[0].first) && (y == t[1].
                second || y == 0 || y + t[2].second == t[0].second);
            if (!ar) continue;
            for (int i = x; i < x + t[2].first; i++) {
                for (int j = y; j < y + t[2].second; j++) {
                    target[i][j] = 1;
                }
            }
            dfsix();
            for (int i = x; i < x + t[2].first; i++) {
                for (int j = y; j < y + t[2].second; j++) {
                    target[i][j] = 0;
                }
            }
        }
    }
}
void check() {
    add(t[0].first , t[0].second);
    int x = t[0].first , y = t[0].second;

    if (t[1].first <= x && t[1].second <= y) {
        check2();
        swap(t[2].first , t[2].second);
        check2();
        swap(t[2].first , t[2].second);
    }
    swap(t[1].first , t[1].second);

    if (t[1].first <= x && t[1].second <= y) {
        check2();
        swap(t[2].first , t[2].second);
        check2();
        swap(t[2].first , t[2].second);
    }

    swap(t[1].first , t[1].second);

    swap(t[0].first , t[0].second);

    x = t[0].first , y = t[0].second;

    if (t[1].first <= x && t[1].second <= y) {
        check2();
        swap(t[2].first , t[2].second);
        check2();
        swap(t[2].first , t[2].second);
    }
    swap(t[1].first , t[1].second);

    if (t[1].first <= x && t[1].second <= y) {
        check2();
        swap(t[2].first , t[2].second);
        check2();
        swap(t[2].first , t[2].second);
    }
}

```

```

        swap(t[1].first, t[1].second);

        swap(t[0].first, t[0].second);
    }

    signed main() {
        int x, y;
        cin >> x >> y;
        if (x > y) swap(x, y);
        t.push_back({x, y});
        cin >> x >> y;
        if (x > y) swap(x, y);
        t.push_back({x, y});
        cin >> x >> y;
        if (x > y) swap(x, y);
        t.push_back({x, y});
        std::sort(t.begin(), t.end());

        do{
            check();
        } while (next_permutation(t.begin(), t.end()));
        // cout << " amogus\n";
        for (auto [x, y]: ans) {
            cout << x << ' ' << y << '\n';
        }
    }
}

```

Task D ()

```
#include <bits/stdc++.h>
using namespace std;
signed main() {
    int n; cin >> n;
    vector<int> maxx(n);
    for (int i = 0; i < n; i++) cin >> maxx[i];
    vector<bool> used(n, 0);
    vector<int> left(n);
    for (int i = 0; i < n; i++) left[i] = maxx[i];
    int grandi_all = 0;
    vector<int> gr_now(n);
    for (int i = 0; i < n; i++) {
        gr_now[i] = left[i] + 1;
        grandi_all = grandi_all ^ gr_now[i];
    }
    if (grandi_all == 0) {
        cout << "-1_-1" << endl;
        cout.flush();
        return 0;
    }
    while (1) {
        if (grandi_all == 0) {
            cout << "-1_-1" << endl;
            cout.flush();
            return 0;
        }
        bool found = 0;

        for (int i = 0; i < n; i++) {
            int need = grandi_all ^ gr_now[i];
            int to_rem = -1;
            if (used[i]) {
                if (need < left[i]) {
                    to_rem = left[i] - need;
                }
            } else {
                if (need == maxx[i]) {
                    to_rem = 0;
                } else {
                    if (need < left[i]) {
                        to_rem = left[i] - need;
                    }
                }
            }

            if (to_rem == -1) continue;
            cout << i + 1 << ' ' << to_rem << endl;
            cout.flush();
            if (to_rem == 0) {
                used[i] = 1;
                left[i] = maxx[i];
            } else {
                left[i] -= to_rem;
            }
            grandi_all = 0;
            gr_now[i] = need;
            found = 1;
            break;
        }
        if (!found) {
            cout << -1 << ' ' << -1 << endl;
            cout.flush();
            return 0;
        }
        int f1, f2;
        cin >> f1 >> f2;
        if (f1 == -1) break;
        grandi_all = grandi_all ^ gr_now[f1 - 1];
        if (f2 == 0) {
            left[f1 - 1] = maxx[f1 - 1];
            used[f1 - 1] = 1;
        } else {
            left[f1 - 1] -= f2;
        }
    }
}
```

```

    }
    gr_now[f1 - 1] = left[f1 - 1];
    grandi_all = grandi_all ^ gr_now[f1 - 1];
}

```

Task E ()

```
#include <bits/stdc++.h>
using namespace std;
int dp[20][20];
void code() {
    int n; cin >> n;
    //    n--;
    vector<int> code;
    code.push_back(0);
    for (int i = 1; i <= 10; i++) {
        for (int diff = 0; diff <= (10 - code.back()); diff++) {

            int what_we_lose = 0;
            for (int j = 0; j <= 10 - code.back() - diff; j++) what_we_lose += dp[10 - i][j];
            if (n > what_we_lose) {
                n -= what_we_lose;
                continue;
            } else {
                code.push_back(code.back() + diff);
                break;
            }
        }
    }
    code.erase(code.begin());

    for (int i = 0; i < 10; i++) {
        for (int x = 0; x < code[i]; x++) cout << 1;
        for (int x = code[i]; x < 10; x++) cout << 0;
        cout << '\n';
    }
    cout << '\n';
}

int get() {
    vector<vector<bool>>> sht(10);
    vector<int> ccnt;
    for (int i = 0; i < 10; i++) {
        string s; cin >> s;
        int c = 0;
        for (int j = 0; j < 10; j++) {
            sht[i].push_back(s[j] == '1');
            c += sht[i].back();
        }
        ccnt.push_back(c);
    }
    std::sort(ccnt.begin(), ccnt.end());
    vector<int> code;
    for (int i: ccnt) code.push_back(i);

    int n = 0;
    for (int i = 0; i < code[0]; i++) {
        int what_we_lose = 0;
        for (int x = 0; x <= 10 - i; x++) what_we_lose += dp[9][x];
        n += what_we_lose;
    }
    for (int i = 1; i < 10; i++) {
        int h = code[i] - code[i - 1];
        for (int c = 0; c < h; c++) {
            int what_we_lose = 0;
            for (int x = 0; x <= 10 - code[i - 1] - c; x++) what_we_lose += dp[10 - i - 1][x];
            n += what_we_lose;
        }
    }
    //    cout << n << " after\n";
}

return n + 1;
}

signed main() {
    dp[0][0] = 1;
    for (int hei = 0; hei < 20; hei++) {
        dp[1][hei] = 1;
    }
}
```



```

for (int l = 2; l <= 10; l++) {
    for (int last_hei = 0; last_hei < 20; last_hei++) {
        for (int nxt_hei = last_hei; nxt_hei < 20; nxt_hei++) {
            dp[l][nxt_hei] += dp[l - 1][last_hei];
        }
    }
    // for (int c = 1; c <= 10; c++) dp[l][c] += dp[l][c - 1];
}

int t; cin >> t;
string s; cin >> s;
while (t--) {
    if (s == "transmit") code();
    else cout << get() << '\n';
}
}

```

Task F ()

```
def convert_to_int(a):
    res = ''
    ptr = 0
    while (ptr < len(a)):
        if a[ptr] == '(':
            p3 = ptr
            while a[p3] != '|':
                p3 += 1
            p2 = p3
            while a[p2] != ')':
                p2 += 1
            res += a[ptr + 1:p3] * int(a[p3 + 1:p2])
            ptr = p2 + 1
            continue
        res += a[ptr]
        ptr += 1
    return int(res)

a = input()
b = input()

a_int = convert_to_int(a)
b_int = convert_to_int(b)

print(a_int + b_int)
```