

# Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	10	52	10	372

## Task A ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

// #define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

int32_t main() {
#ifdef __APPLE__
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
    array<int, 6> ar{};
    for (auto &i : ar) cin >> i;
    vector<int> p(6);
    iota(all(p), 0);
    do {
        bool f = true;
        for (int i = 0; i < 6; ++i) {
            int t = 0;
            for (int j = 0; j < i; ++j) {
                if (p[i] > p[j]) {
                    ++t;
                }
            }
            if (t != ar[i] - 1) {
                f = false;
                break;
            }
        }
        if (f) {
            for (auto i : p) {
                cout << i + 1 << " ";
            }
            exit(0);
        }
    } while (next_permutation(all(p)));
    return 0;
}
```

## Task B ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

#define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

const int MX = 1e5;

void solveFirst() {
    int n;
    cin >> n;
    vector<int> v(n);
    for (auto &i : v) {
        cin >> i;
    }
    cout << (int)accumulate(all(v), 0ll) * MX;
}

void solveSecond() {
    int n;
    cin >> n;
    vector<int> v(n);
    for (auto &i : v) {
        cin >> i;
    }
    int sumOld = v[0] / MX;
    int sumNew = 0;
    for (int i = 0; i < n; ++i) {
        sumNew += (v[i]) % MX;
    }
    cout << sumOld + sumNew;
}

int32_t main() {
#ifdef __APPLE__
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
    string s;
    cin >> s;
    if (s == "first") {
        solveFirst();
    } else {
        solveSecond();
    }
    return 0;
}
```

## Task C ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

#define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

bool isIn(int xp, int yp, int x1, int y1, int x2, int y2) {
    return (xp > x1 && xp < x2 && yp > y1 && yp < y2);
}

bool intersect(int x1, int y1, int x2, int y2,
               int x3, int y3, int x4, int y4) {
    return (
        isIn(x1, y1, x3, y3, x4, y4) ||
        isIn(x2, y2, x3, y3, x4, y4)
    );
}

int intersect(int x1, int x2,
              int x3, int x4) {
    if (x1 >= x3 && x1 <= x4) {
        return min(x2, x4) - x1;
    }
    if (x2 >= x3 && x2 <= x4) {
        return x2 - max(x1, x3);
    }
    if (x3 >= x1 && x3 <= x2) {
        return x4 - x3;
    }
    return 0;
}

bool contactHor(int x1, int y1, int x2, int y2,
                int x3, int y3, int x4, int y4) {
    /* [----]
     * [rect][    ]
     * [----][rect]
     */
    if (x2 == x3 || x4 == x1 || x2 == x4 || x1 == x3) {
        return intersect(y1, y2, y3, y4);
    }
    return false;
}

bool contactVer(int x1, int y1, int x2, int y2,
                 int x3, int y3, int x4, int y4) {
    if (y2 == y3 || y4 == y1 || y1 == y3 || y2 == y4) {
        return intersect(x1, x2, x3, x4);
    }
    return false;
}

int32_t main() {
#ifdef _APPLE_
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
}
```

```

array <pii, 3> rect{};
int X = 0;
int Y = 0;
for (auto& [x, y] : rect) {
    cin >> x >> y;
    X = max(x, X);
    Y = max(y, Y);
}
int n = 3;
vector <pii> ans;
ans.insert(ans.begin(), rect.begin(), rect.end());
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
        if (i == j) {
            continue;
        }
        if (rect[i].first == rect[j].first && rect[j].second > rect[i].second) {
            ans.emplace_back(rect[j].second - rect[i].second, rect[i].first);
        }
        if (rect[i].second == rect[j].second && rect[j].first > rect[i].first) {
            ans.emplace_back(rect[j].first - rect[i].first, rect[i].second);
        }
        swap(rect[i].first, rect[i].second);
        if (rect[i].first == rect[j].first && rect[j].second > rect[i].second) {
            ans.emplace_back(rect[j].second - rect[i].second, rect[i].first);
        }
        if (rect[i].second == rect[j].second && rect[j].first > rect[i].first) {
            ans.emplace_back(rect[j].first - rect[i].first, rect[i].second);
        }
        swap(rect[i].first, rect[i].second);
    }
}
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < n; ++j) {
        if (i == j) {
            continue;
        }
        for (int k = 0; k < n; ++k) {
            if (k == i || k == j) {
                continue;
            }
            // k - big
            for (int rotMask = 0; rotMask < 8; ++rotMask) {
                vector <pii> r = {rect[i], rect[j], rect[k]};
                for (int t = 0; t < 3; ++t) {
                    if (rotMask >> t & 1) {
                        swap(r[t].first, r[t].second);
                    }
                }
                if (r[2].second - r[0].second == r[1].second && r[1].first == r[2].first - r[0].first) {
                    ans.emplace_back(r[0].first, r[1].second);
                    ans.emplace_back(r[2].first - r[0].first, r[2].second - r[1].second);
                }
                if (r[1].first == r[2].first && r[0].first == r[2].first) {
                    ans.emplace_back(r[1].first, r[2].second - r[0].second - r[1].second);
                }
                if (r[1].second == r[0].second && r[0].first + r[1].first == r[2].first) {
                    ans.emplace_back(r[2].second - r[1].second, r[2].first);
                }
                if (r[0].first == r[2].first && r[0].second + r[1].second == r[2].second) {
                    ans.emplace_back(r[2].second - r[0].second, r[2].first - r[1].first);
                }
                if (r[0].first + r[1].first > r[2].first) {
                    continue;
                }
                if (max(r[0].second, r[1].second) > r[2].second) {
                    continue;
                }
                if (r[1].second > r[2].second - r[0].second) {
                    ans.emplace_back(r[2].second - r[0].second, r[0].first);
                    if (r[1].second == r[2].second) {
                        ans.emplace_back(r[2].first - r[1].first - r[0].first, r[2].second);
                    }
                }
            }
        }
    }
}

```

```

    }
}
}
}

/*
 * #####
 * #..***#
 * #..*  *#
 * #%%***#
 * #%%...#
 * #%%...#
 * #####
 */
set <pii> res;
for (auto [x, y] : ans) {
    res.insert({min(x, y), max(x, y)});
}
for (auto [x, y] : res) {
    if (min(x, y) > 0) {
        cout << x << "_" << y << "\n";
    }
}
return 0;
}

```

## Task D ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

#define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

// C GOVNO

int v[50];

bool getQ(pii &t) {
    int a, b;
    cin >> a >> b;
    t = {a, b};
    if (a == -1) {
        exit(0);
    }
    return true;
}

int n;

const int MX = 51;
int dp[8][MX][MX][MX];
int mv[8][MX][MX][MX];

int calc(int mask, int a, int b, int c) {
    if (dp[mask][a][b][c]) {
        return dp[mask][a][b][c];
    }
    if (mask & 1) {
        if (calc(mask ^ 1, v[0], b, c) == 1) {
            mv[mask][a][b][c] = 0;
            return dp[mask][a][b][c] = 2;
        }
    }
    if (mask & 2) {
        if (calc(mask ^ 2, a, v[1], c) == 1) {
            mv[mask][a][b][c] = 1;
            return dp[mask][a][b][c] = 2;
        }
    }
    if (mask & 4) {
        if (calc(mask ^ 4, a, b, v[2]) == 1) {
            mv[mask][a][b][c] = 2;
            return dp[mask][a][b][c] = 2;
        }
    }
    for (int a1 = 0; a1 < a; ++a1) {
        if (calc(mask, a1, b, c) == 1) {
            mv[mask][a][b][c] = a1 + 500;
            return dp[mask][a][b][c] = 2;
        }
    }
    for (int b1 = 0; b1 < b; ++b1) {
        if (calc(mask, a, b1, c) == 1) {
            mv[mask][a][b][c] = b1 + 1000;
            return dp[mask][a][b][c] = 2;
        }
    }
}
```

```

    for (int c1 = 0; c1 < c; ++c1) {
        if (calc(mask, a, b, c1) == 1) {
            mv[mask][a][b][c] = c1 + 1500;
            return dp[mask][a][b][c] = 2;
        }
    }
    if (a) {
        mv[mask][a][b][c] = 500;
    } else if (b) {
        mv[mask][a][b][c] = 1000;
    } else if (c) {
        mv[mask][a][b][c] = 1500;
    } else if (mask & 1) {
        mv[mask][a][b][c] = 0;
    } else if (mask & 2) {
        mv[mask][a][b][c] = 1;
    } else {
        mv[mask][a][b][c] = 2;
    }
    return dp[mask][a][b][c] = 1;
}

int32_t main() {
#ifdef __APPLE__
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
    cin >> n;
    for (int i = 0; i < n; ++i) {
        cin >> v[i];
    }
    pii val{};
    dp[0][0][0][0] = 1;
    int curm = (1 << n) - 1;
    int cura = v[0], curb = v[1], curc = v[2];
    int pr = -1;
    while (true) {
        cerr << curm << " " << cura << " " << curb << " " << curc << endl;
        int dpval = calc(curm, cura, curb, curc);
        if (pr != -1 && dpval == 1) {
            assert(false);
        }
        if (pr == -1) {
            pr = dpval;
        }
        if (dpval == 1) {
            cout << "-1_-1" << endl;
            getQ(val);
            exit(0);
        }
        int mov = mv[curm][cura][curb][curc];
        if (mov >= 1500) {
            cout << "3_" << curc - mov + 1500 << endl;
            curc = mov - 1500;
        } else if (mov >= 1000) {
            cout << "2_" << curb - mov + 1000 << endl;
            curb = mov - 1000;
        } else if (mov >= 500) {
            cout << "1_" << cura - mov + 500 << endl;
            cura = mov - 500;
        } else {
            curm ^= (1 << mov);
            if (mov == 0) {
                cura = v[0];
            } else if (mov == 1) {
                curb = v[1];
            } else {
                curc = v[2];
            }
            cout << (mov + 1) << "_0" << endl;
        }
    }
}

```

```

getQ(val);
if (val.second == 0) {
    if (val.first == 1) {
        cura = v[0];
    } else if (val.first == 2) {
        curb = v[1];
    } else {
        curc = v[2];
    }
    curm ^= (1 << (val.first - 1));
} else {
    if (val.first == 1) {
        cura ^= val.second;
    } else if (val.first == 2) {
        curb ^= val.second;
    } else {
        curc ^= val.second;
    }
}
}
return 0;
}

```



## Task E ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

#define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

// C GOVNO

//2222222222233
//4444444444445

const int N = 10;

void solveFirst() {
    int n;
    cin >> n;
    mt19937 rnd(chrono::high_resolution_clock::now().time_since_epoch().count());
    array<string, N> res{};
    for (int i = 0; i < N; ++i) {
        res[i] = string(N, '0');
    }
    int ptr = 0;
    for (int i = 0; i < N; ++i) {
        if (n >> i & 1) {
            for (int j = 0; j < min((i + 1), N); ++j) {
                res[ptr][j] = '1';
            }
            ++ptr;
        }
    }
    if ((n >> N & 1) && ptr == N - 1) {
        fill(all(res[ptr]), '1');
        for (int i = 0; i < N; ++i) {
            shuffle(all(res[i]), rnd);
        }
    }
    for (auto& i : res) {
        cout << i << "\n";
    }
}

void solveSecond() {
    array<string, N> res;
    for (auto& i : res) {
        cin >> i;
    }
    int p = 0;
    vector<int> cnt(N + 1);
    for (int i = 0; i < N; ++i) {
        int t = count(all(res[i]), '1');
        if (t) {
            ++cnt[t - 1];
        }
    }
    bool f = true;
    for (int j = 0; j < N; ++j) {
        int c = 0;
        for (int i = 0; i < N; ++i) {
            c += (res[i][j] - '0');
        }
    }
}
```

```

        if (c == 1) {
            f = false;
            break;
        }
    }
    if (f) {
        cout << 1535;
        return;
    }
    for (int i = 0; i <= N; ++i) {
        if (cnt[i]) {
            p |= (1 << i);
        }
    }
    cout << p;
}

int32_t main() {
#ifdef __APPLE__
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
    int t;
    cin >> t;
    string s;
    cin >> s;
    while (t--) {
        if (s == "transmit") {
            solveFirst();
        } else {
            solveSecond();
        }
        cout << "\n";
    }
    return 0;
}

```

## Task F ()

```
#include <iostream>
#include <set>
#include <vector>
#include <map>
#include <algorithm>
#include <cmath>
#include <iomanip>
#include <chrono>
#include <random>
#include <cassert>
#include <array>

using namespace std;

#define int long long
using pii = pair<int, int>;
#define all(x) (x).begin(), (x).end()

// C GOVNO

//2222222222233
//4444444444445

int32_t main() {
#ifdef __APPLE__
    freopen("in.txt", "r", stdin);
    freopen("out.txt", "w", stdout);
#else
    ios_base::sync_with_stdio(false);
    cin.tie(0);
#endif
    srand(228);
    // while (true) {
    //     int aaa = rand() % 100;
    //     int bbb = rand() % 100;
    string a, b;
    //     a = to_string(aaa);
    //     b = to_string(bbb);
    cin >> a >> b;
    vector<pii> a1;
    vector<pii> b1;
    int n = (int) a.size();
    int m = (int) b.size();
    string cur;
    int curstate = 0;
    pii val;
    for (int i = 0; i < n; ++i) {
        if (a[i] < '0' || a[i] > '9') {
            if (a[i] == '(') {
                curstate = 1;
            } else if (a[i] == '|') {
                curstate = 2;
            } else {
                curstate = 0;
            }
            if (curstate == 2) {
                val.first = stoi(cur);
            } else if (curstate == 0) {
                val.second = stoi(cur);
                a1.emplace_back(val);
            }
            cur = "";
        } else {
            cur += a[i];
            if (curstate == 0) {
                a1.emplace_back(a[i] - '0', 1);
                cur = "";
            }
        }
    }
    for (int i = 0; i < m; ++i) {
        if (b[i] < '0' || b[i] > '9') {
```

```

        if (b[i] == '(') {
            curstate = 1;
        } else if (b[i] == '|') {
            curstate = 2;
        } else {
            curstate = 0;
        }
        if (curstate == 2) {
            val.first = stoi(cur);
        } else if (curstate == 0) {
            val.second = stoi(cur);
            b1.emplace_back(val);
        }
        cur = "";
    } else {
        cur += b[i];
        if (curstate == 0) {
            b1.emplace_back(b[i] - '0', 1);
            cur = "";
        }
    }
}
int sumlen1 = 0, sumlen2 = 0;
for (auto [x, y]: a1) {
    sumlen1 += y;
}
for (auto [x, y]: b1) {
    sumlen2 += y;
}
reverse(a1.begin(), a1.end());
reverse(b1.begin(), b1.end());
if (sumlen1 < sumlen2) {
    a1.emplace_back(0, sumlen2 - sumlen1);
}
if (sumlen1 > sumlen2) {
    b1.emplace_back(0, sumlen1 - sumlen2);
}
vector<int> xs;
{
    int x = 0;
    for (auto [t, i]: a1) {
        x += i;
        xs.push_back(x);
    }
    x = 0;
    for (auto [t, i]: b1) {
        x += i;
        xs.push_back(x);
    }
    sort(all(xs));
    xs.resize(unique(all(xs)) - xs.begin());
}
vector<pii> ans;
vector<pii> a2, b2;
int curi = 0;
int curx = 0;
int pr = 0;
for (auto x: xs) {
    while (curi < (int) a1.size() && curx + a1[curi].second < x) {
        curx += a1[curi].second;
        ++curi;
    }
    a2.emplace_back(a1[curi].first, x - pr);
    pr = x;
}
curi = 0;
curx = 0;
pr = 0;
for (auto x: xs) {
    while (curi < (int) b1.size() && curx + b1[curi].second < x) {
        curx += b1[curi].second;
        ++curi;
    }
    b2.emplace_back(b1[curi].first, x - pr);
}

```

```

        pr = x;
    }
    n = (int) b2.size();
    int add = 0;
    for (int i = 0; i < n; ++i) {
        auto [v1, x1] = a2[i];
        auto [v2, x2] = b2[i];
        if (v1 > v2) {
            swap(v1, v2);
        }
        int nadd = 0;
        if (v2 == 9 && add) {
            nadd = 1;
            ans.emplace_back(v1, x2);
        } else {
            if (v1 + v2 == 9 && add) {
                nadd = 1;
                ans.emplace_back(0, x1);
            } else if (v1 + v2 <= 9) {
                if (add) {
                    ans.emplace_back(v1 + v2 + add, 1);
                    ans.emplace_back(v1 + v2, x1 - 1);
                } else {
                    ans.emplace_back(v1 + v2, x1);
                }
            } else {
                nadd = 1;
                ans.emplace_back((v1 + v2 + add) % 10, 1);
                ans.emplace_back((v1 + v2 + 1) % 10, x1 - 1);
            }
        }
        add = nadd;
    }
    if (add) {
        ans.emplace_back(1, 1);
    }
    vector<pii> ans1;
    for (auto [x, y]: ans) {
        if (y) {
            ans1.emplace_back(x, y);
        }
    }
    while (!ans1.empty() && ans1.back().first == 0) {
        ans1.pop_back();
    }
    if (ans1.empty()) {
        cout << 0;
        return 0;
    }
    reverse(all(ans1));
    for (auto [x, y]: ans1) {
        cout << "(" << x << "|" << y << ")";
    }

    // string prt;
    // for (auto [x, y] : ans1) {
    //     while (y--) {
    //         prt += (char) (x + '0');
    //     }
    // }
    // if (stoi(prt) != aaa + bbb) {
    //     cerr << " " << aaa << " " << bbb << "\n";
    //     return 0;
    // }
    // }
    // }
    return 0;
}

```