

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	48	10	458

Task A ()

```
// clang-format off

// #pragma GCC target ("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2")
// #pragma GCC optimize("Ofast")
// #pragma GCC optimize("no-stack-protector")
// #pragma GCC optimize("unroll-loops")
// #pragma GCC optimize("unswitch-loops")
// #pragma GCC optimize("fast-math")
// #pragma GCC optimize("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":\n" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:\n" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    int n = 6;
    vector<int> a(6);
```

```

for (int &i: a)
    cin >> i, i--;
vector<int> res(n);
iota(all(res), 0);
do {
    bool ok = true;
    for (int i = 0; i < n; i++) {
        int pos = 0;
        for (int j = 0; j < i; j++)
            if (res[j] < res[i])
                pos++;
        if (pos != a[i])
            ok = false;
    }
    if (ok) {
        for (int i: res) cout << i + 1 << " "; cout << '\n';
        return;
    }
} while (next_permutation(all(res)));
}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```

Task B ()

```
// clang-format off

// #pragma GCC target ("sse , sse2 , sse3 , ssse3 , sse4 , popcnt , abm , mmx , avx , avx2 ")
// #pragma GCC optimize ("Ofast")
// #pragma GCC optimize ("no-stack-protector")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC optimize ("unswitch-loops")
// #pragma GCC optimize ("fast-math")
// #pragma GCC optimize ("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":_" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:_" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    string s;
    cin >> s;
    if (s == "first") {
        int n;
        cin >> n;
        int x = 0;
        for (int i = 0; i < n; i++) {
            int a;
            cin >> a;
            x += a;
        }
        cout << x * 1000 << '\n';
    } else {
        int n;
        cin >> n;
        int x = 0;
        for (int i = 0; i < n; i++) {
            int a;
```

```

        cin >> a;
        if (i == 0) x += a / 1000;
        x += a % 1000;
    }
    cout << x << '\n';
}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```

Task C ()

```
// clang-format off

// #pragma GCC target ("sse, sse2, sse3, ssse3, sse4, popcnt, abm, mmx, avx, avx2")
// #pragma GCC optimize ("Ofast")
// #pragma GCC optimize ("no-stack-protector")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC optimize ("unswitch-loops")
// #pragma GCC optimize ("fast-math")
// #pragma GCC optimize ("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":_" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:_" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    int n = 3;
    vector<int> aa(n), bb(n);
    for (int i = 0; i < n; i++)
        cin >> aa[i] >> bb[i];
    vector<int> p{0, 1, 2};
    vector<pair<int, int>> res;
    for (int i = 0; i < n; i++)
        res.push_back({aa[i], bb[i]});
    for (int mask = 0; mask < 1 << n; mask++) {
        for (int i = 0; i < n; i++) if (mask >> i & 1) swap(aa[i], bb[i]);
        do {
            vector<int> a(n), b(n);
            for (int i = 0; i < n; i++)
                a[i] = aa[p[i]], b[i] = bb[p[i]];
            if (a[0] == a[1] and b[0] > b[1]) res.push_back({a[0], b[0] - b[1]});
            if (a[1] + a[2] > a[0] and b[1] + b[2] > b[0])
                continue;
            if (a[1] + a[2] > a[0] and max(a[1], a[2]) <= a[0] and b[1] + b[2] <= b
```

```

        [0])
        res.push_back({a[0] - a[1], b[1]});
    if (a[0] == a[1] and b[0] > b[1]) res.push_back({a[0], b[0] - b[1]});
    if (a[1] + a[2] > a[0] and b[1] + b[2] > b[0])
        continue;
    if (b[1] + b[2] == b[0] and a[1] == a[0]) res.push_back({a[0] - a[2], b
        [2]});
    if (a[1] + a[2] == a[0]) {
        if (b[1] + b[2] == b[0])
            res.push_back({a[1], b[2]}), res.push_back({a[2], b[1]});
        if (b[1] == b[2])
            res.push_back({a[0], b[0] - b[1]});
    }
    if (max(a[1], a[2]) <= a[0] and a[1] + a[2] > a[0] and b[1] + b[2] <= b
        [0])
        res.push_back({a[0] - a[1], b[1]});
    if (a[1] + a[2] < a[0] and b[2] == b[0])
        res.push_back({a[0] - a[1] - a[2], b[0]});
    } while (next_permutation(all(p)));
    for (int i = 0; i < n; i++) if (mask >> i & 1) swap(aa[i], bb[i]);
}
for (auto&[i, j] : res) if (i > j) swap(i, j);
auto cur = res;
res.clear();
for (auto [i, j] : cur)
    if (min(i, j) > 0)
        res.push_back({i, j});
sort(all(res));
res.resize(unique(all(res)) - res.begin());
for (auto [i, j] : res)
    cout << i << " " << j << "\n";
}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```

Task D ()

```
// clang-format off

// #pragma GCC target ("sse , sse2 , sse3 , ssse3 , sse4 , popcnt , abm , mmx , avx , avx2 ")
// #pragma GCC optimize ("Ofast")
// #pragma GCC optimize ("no-stack-protector")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC optimize ("unswitch-loops")
// #pragma GCC optimize ("fast-math")
// #pragma GCC optimize ("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":\n" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:\n" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

const int N = 51;

void solve() {
    int n;
    cin >> n;
    vector<int> a(n);
    for (int &i: a)
        cin >> i;
    if (n == 1) {
        cout << 1 << "\n" << a[0] << endl;
        int x, y;
        cin >> x >> y;
        if (x == -1) return;
        cout << 1 << "\n" << a[0] << endl;
    } else if (n == 2) {
        static int dp[4][N][N];
        for (int mask = 0; mask < 4; mask++) {
            for (int i = 0; i <= a[0]; i++) {
                for (int j = 0; j <= a[1]; j++) {
                    if (mask & 1) dp[mask][i][j] |= !dp[mask ^ 1][a[0]][j];
                }
            }
        }
    }
}
```

```

        if (mask & 2) dp[mask][i][j] |= !dp[mask ^ 2][i][a[1]];
        for (int k = 1; k <= i; k++)
            dp[mask][i][j] |= !dp[mask][i - k][j];
        for (int k = 1; k <= j; k++)
            dp[mask][i][j] |= !dp[mask][i][j - k];
    }
}
if (!dp[3][a[0]][a[1]]) return (void) (cout << "-1_-1" << endl);
int x, y;
int mask = 3;
auto b = a;
while (1) {
    if (mask & 1 and !dp[mask ^ 1][a[0]][b[1]]) {
        cout << 1 << "_" << 0 << endl;
        mask ^= 1;
        b[0] = a[0];
    } else if (mask & 2 and !dp[mask ^ 2][b[0]][a[1]]) {
        cout << 2 << "_" << 0 << endl;
        mask ^= 2;
        b[1] = a[1];
    } else {
        for (int k = 0; k < N; k++) {
            if (b[0] >= k and !dp[mask][b[0] - k][b[1]]) {
                cout << 1 << "_" << k << endl;
                b[0] -= k;
                break;
            }
            if (b[1] >= k and !dp[mask][b[0]][b[1] - k]) {
                cout << 2 << "_" << k << endl;
                b[1] -= k;
                break;
            }
        }
    }
    cin >> x >> y, x--;
    if (x == -2) break;
    if (y == 0) b[x] = a[x], mask ^= (1 << x);
    else b[x] -= y;
}
} else if (n == 3) {
    static int dp[8][N][N][N];
    for (int mask = 0; mask < 8; mask++) {
        for (int i = 0; i <= a[0]; i++) {
            for (int j = 0; j <= a[1]; j++) {
                for (int k = 0; k <= a[2]; k++) {
                    if (mask & 1) dp[mask][i][j][k] |= !dp[mask ^ 1][a[0]][j][k];
                    if (mask & 2) dp[mask][i][j][k] |= !dp[mask ^ 2][i][a[1]][k];
                    if (mask & 4) dp[mask][i][j][k] |= !dp[mask ^ 4][i][j][a[2]];
                    for (int l = 1; l <= i; l++)
                        dp[mask][i][j][k] |= !dp[mask][i - l][j][k];
                    for (int l = 1; l <= j; l++)
                        dp[mask][i][j][k] |= !dp[mask][i][j - l][k];
                    for (int l = 1; l <= k; l++)
                        dp[mask][i][j][k] |= !dp[mask][i][j][k - l];
                }
            }
        }
    }
}
if (!dp[7][a[0]][a[1]][a[2]]) return (void) (cout << "-1_-1" << endl);
int x, y;
int mask = 7;
auto b = a;
while (1) {
    if (mask & 1 and !dp[mask ^ 1][a[0]][b[1]][b[2]]) {
        mask ^= 1;
        cout << 1 << "_" << 0 << endl;
        b[0] = a[0];
    }
}

```



```

    } else if (mask & 2 and !dp[mask ^ 2][b[0]][a[1]][b[2]]) {
        mask ^= 2;
        cout << 2 << "┘" << 0 << endl;
        b[1] = a[1];
    } else if (mask & 4 and !dp[mask ^ 4][b[0]][b[1]][a[2]]) {
        mask ^= 4;
        cout << 3 << "┘" << 0 << endl;
        b[2] = a[2];
    } else {
        for (int k = 0; k < N; k++) {
            if (b[0] >= k and !dp[mask][b[0] - k][b[1]][b[2]]) {
                cout << 1 << "┘" << k << endl;
                b[0] -= k;
                break;
            }
            if (b[1] >= k and !dp[mask][b[0]][b[1] - k][b[2]]) {
                cout << 2 << "┘" << k << endl;
                b[1] -= k;
                break;
            }
            if (b[2] >= k and !dp[mask][b[0]][b[1]][b[2] - k]) {
                cout << 3 << "┘" << k << endl;
                b[2] -= k;
                break;
            }
        }
    }
    cin >> x >> y, x--;
    if (x == -2) break;
    if (y == 0) b[x] = a[x], mask ^= (1 << x);
    else b[x] -= y;
}

}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```

Task E ()

```
// clang-format off

// #pragma GCC target ("sse , sse2 , sse3 , ssse3 , sse4 , popcnt , abm , mmx , avx , avx2 ")
// #pragma GCC optimize ("Ofast")
// #pragma GCC optimize ("no-stack-protector")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC optimize ("unswitch-loops")
// #pragma GCC optimize ("fast-math")
// #pragma GCC optimize ("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

#define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":_" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:_" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    int n = 10;
    int t;
    cin >> t;
    string op;
    cin >> op;
    while (t--) {
        if (op == "transmit") {
            int x;
            cin >> x;
            vector<vector<int>>> a(n, vector<int>(n, 0));
            for (int i = 0; i < n; i++) {
                if (x >> i & 1) {
                    for (int j = 0; j < i + 1; j++) a[i][j] = 1;
                }
            }
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    cout << a[i][j];
                }
            }
        }
    }
}
```

```

        }
        cout << '\n';
    }
    cout << '\n';
} else {
    int res = 0;
    char a;
    for (int i = 0; i < n; i++) {
        int d = 0;
        for (int j = 0; j < n; j++) {
            cin >> a;
            d += a - '0';
        }
        d--;
        if (d >= 0) res |= 1 << d;
    }
    cout << res << '\n';
}

}

}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```

Task F ()

```
// clang-format off

// #pragma GCC target ("sse , sse2 , sse3 , ssse3 , sse4 , popcnt , abm , mmx , avx , avx2 ")
// #pragma GCC optimize ("Ofast")
// #pragma GCC optimize ("no-stack-protector")
// #pragma GCC optimize ("unroll-loops")
// #pragma GCC optimize ("unswitch-loops")
// #pragma GCC optimize ("fast-math")
// #pragma GCC optimize ("rename-registers")

#include <iostream>
#include <vector>
#include <algorithm>
#include <queue>
#include <cmath>
#include <set>
#include <stack>
#include <bitset>
#include <map>
#include <ctime>
#include <numeric>
#include <random>
#include <cassert>
#include <cstring>
#include <chrono>
#include <functional>

// #define int long long
#define uint unsigned long long
#define double long double
#define INF (int) 1e18 / 2
#define all(a) a.begin(), a.end()
#define debug(a) cerr << #a << ":_" << a << endl
#define YES return (void) (cout << "Yes\n")
#define NO return (void) (cout << "No\n")
#define ls x << 1
#define rs x << 1 | 1

#ifdef JUSTNIK
#define start cout.setf(ios::fixed); cout.precision(10); int START = clock()
#define finish cout << "\ntime:_" << (clock() - START) / (double)(CLOCKS_PER_SEC); return 0
#else
#define start cin.tie(NULL); cout.tie(NULL); cout.setf(ios::fixed); cout.precision(10); ios_base::
    sync_with_stdio(false)
#define finish return 0
#endif

using namespace std;

mt19937 rnd(chrono::steady_clock::now().time_since_epoch().count());

void solve() {
    string s, t;
    cin >> s >> t;
    auto get = [](string s) {
        int n = s.size();
        string res = "";
        for (int i = 0; i < n; i++) {
            string d = "";
            string k = "";
            if (s[i] == '(') {
                for (i++; s[i] != '|'; i++)
                    d += s[i];
                for (i++; s[i] != ')'; i++)
                    k += s[i];
                int x = stoll(k);
                for (int j = 0; j < x; j++)
                    res += d;
            } else {
                res += s[i];
            }
        }
    };
}
```

```

        }
    }
    return res;
};

s = get(s), t = get(t);
reverse(all(s)), reverse(all(t));
string res = "";
int n = max(s.size(), t.size());
while (s.size() < n) s += '0';
while (t.size() < n) t += '0';
int one = 0;
for (int i = 0; i < n; i++) {
    int x = s[i] - '0', y = t[i] - '0';
    res += (char) ((x + y + one) % 10 + '0');
    one = (x + y + one) / 10;
}
if (one) res += '1';
while (!res.empty() and res.back() == '0') res.pop_back();
if (res == "") res = "0";
reverse(all(res));
n = res.size();
string cur = "";
for (int i = 0; i < n; i++) {
    int j;
    for (j = i; j < n and res[i] == res[j]; j++);
    if (4 + to_string(j - i).size() < j - i) {
        cur += '(';
        cur += res[i];
        cur += '|';
        cur += to_string(j - i);
        cur += ')';
    } else {
        for (; i < j; i++)
            cur += res[i];
    }
    i = j - 1;
}
res = cur;
assert(res.size() <= 100000);
cout << res << '\n';
}

int32_t main() {
    start;

    solve();

    finish;
}

// clang-format on

```