

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	10	28	65	403

Task A ()

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>

using namespace std;

typedef long double ld;

signed main() {
    int n = 6;
    vector<int> cur;
    for (int i = 0; i < n; ++i) {
        int x; cin >> x;
        vector<int> nw;
        for (int j = 0; j < x - 1; ++j)
            nw.push_back(cur[j]);
        nw.push_back(i + 1);
        for (int j = x - 1; j < (int)cur.size(); ++j)
            nw.push_back(cur[j]);
        cur = nw;
    }
    vector<int> res(n);
    for (int i = 0; i < n; ++i)
        res[cur[i] - 1] = i + 1;
    for (auto &x : res)
        cout << x << ' ';
    cout << '\n';
}
```


Task B ()

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#define int long long
using namespace std;

const int mod = 1e9 + 7;

int bp(int a, int n) {
    if (n == 0)
        return 1;
    if (n & 1) {
        return (a * bp(a, n - 1)) % mod;
    }
    int t = bp(a, n / 2);
    return (t * t) % mod;
}

int dv(int a, int b) {
    return (a * bp(b, mod - 2)) % mod;
}

void encode() {
    int n; cin >> n;
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        int x; cin >> x;
        sum += x;
    }
    sum = dv(sum, n);
    cout << sum << '\n';
}

void decode() {
    int n; cin >> n;
    int sum = 0;
    for (int i = 0; i < n; ++i) {
        int x; cin >> x;
        sum += x;
    }

    sum %= mod;
    cout << sum << '\n';
}

signed main() {
    string s; cin >> s;
    if (s == "first") {
        encode();
    } else {
        decode();
    }
}

/*

first
5
1 2 3 4 5

second
5
9 10 11 12 13
*/
```


Task C ()

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <set>
using namespace std;

struct str {
    int a, b;
    str(int A = 0, int B = 0) {
        a = A; b = B;
    }
    void make() {
        if (a > b)
            swap(a, b);
    }
};

vector<str> v;
set<pair<int, int>> st;
void go(int l1, int r1, int l2, int r2, int l3, int r3) {
    for (int i = l1; i < r1; ++i) {
        for (int j = l2; j < r2; ++j) {
            for (int k = l3; k < r3; ++k) {
                auto a = v[i], b = v[j], c = v[k];
                for (int _ = 0; _ < 2; ++_) {
                    if (b.a == a.a) {
                        st.insert({a.a, a.b - b.b});
                    }
                    if (b.b + c.b <= a.b && a.a >= b.a && c.a > a.a - b.a - (b.b + c.b == a.b)) {
                        st.insert({a.a - c.a, c.b});
                    }
                    if (c.a + b.a == a.a) {
                        if (c.b == b.b) {
                            st.insert({a.a, a.b - b.b});
                        }
                    }
                }
            }
        }
    }
}

signed main() {
    int a1, b1, a2, b2, a3, b3;
    cin >> a1 >> b1 >> a2 >> b2 >> a3 >> b3;
    v.push_back(str(a1, b1));
    v.push_back(str(b1, a1));
    v.push_back(str(a2, b2));
    v.push_back(str(b2, a2));
    v.push_back(str(a3, b3));
    v.push_back(str(b3, a3));

    st.insert({a1, b1});
    st.insert({a2, b2});
    st.insert({a3, b3});
    go(0, 2, 2, 4, 4, 6);
    go(0, 2, 4, 6, 2, 4);

    go(2, 4, 0, 2, 4, 6);
    go(2, 4, 4, 6, 0, 2);

    go(4, 6, 0, 2, 2, 4);
    go(4, 6, 2, 4, 0, 2);

    vector<pair<int, int>> res;
```



```

for (auto &x : st)
    res.push_back(x);
for (auto &x : v)
    res.push_back({x.a, x.b});

for (auto &p : res)
    if (p.first > p.second)
        swap(p.first, p.second);

sort(res.begin(), res.end());
res.resize(unique(res.begin(), res.end()) - res.begin());

for (auto &p : res) {
    if (p.first <= 0)
        continue;
    cout << p.first << ' ' << p.second << '\n';
}
}

```


Task D ()

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <string>
#include <map>
#pragma GCC optimize("Ofast")
using namespace std;

vector<bool> used;
vector<int> mx;

map< pair<vector<bool>, vector<int>> , bool> mp;

bool dp[2][2][2][51][51][51];

pair<int, int> find(int a, int b, int c, int t1, int t2, int t3) {
    if (t1 && dp[0][t2][t3][mx[0]][b][c] == false) {
        return { 1, 0 };
    }
    if (t2 && dp[t1][0][t3][a][mx[1]][c] == false) {
        return { 2, 0 };
    }
    if (t3 && dp[t1][t2][0][a][b][mx[2]] == false) {
        return { 3, 0 };
    }

    for (int x = 1; x <= a; ++x) {
        if (dp[t1][t2][t3][a - x][b][c] == false) {
            return { 1, x };
        }
    }
    for (int x = 1; x <= b; ++x) {
        if (dp[t1][t2][t3][a][b - x][c] == false) {
            return { 2, x };
        }
    }
    for (int x = 1; x <= c; ++x) {
        if (dp[t1][t2][t3][a][b][c - x] == false) {
            return { 3, x };
        }
    }
    return { -1, -1 };
}

signed main() {
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    int n; cin >> n;
    mx.resize(n);
    used.resize(n, 1);
    used.push_back(0);
    used.push_back(0);
    for (auto &x : mx)
        cin >> x;
    mx.push_back(0);
    mx.push_back(0);
    // go();

    for (int t1 = 0; t1 < 2; ++t1) {
        for (int t2 = 0; t2 < 2; ++t2) {
            for (int t3 = 0; t3 < 2; ++t3) {
                for (int a = 0; a <= mx[0]; ++a) {
                    for (int b = 0; b <= mx[1]; ++b) {
                        for (int c = 0; c <= mx[2]; ++c) {
                            if (t1 && dp[0][t2][t3][mx[0]][b][c] == false) {
                                dp[t1][t2][t3][a][b][c] = true;
                                continue;
                            }
                        }
                        if (t2 && dp[t1][0][t3][a][mx[1]][c] == false) {
                            dp[t1][t2][t3][a][b][c] = true;
                            continue;
                        }
                    }
                }
            }
        }
    }
```



```

    }
    if (t3 && dp[t1][t2][0][a][b][mx[2]] == false) {
        dp[t1][t2][t3][a][b][c] = true;
        continue;
    }

    for (int x = 1; x <= a; ++x) {
        if (dp[t1][t2][t3][a - x][b][c] == false) {
            dp[t1][t2][t3][a][b][c] = true;
            break;
        }
    }
    for (int x = 1; x <= b; ++x) {
        if (dp[t1][t2][t3][a][b - x][c] == false) {
            dp[t1][t2][t3][a][b][c] = true;
            break;
        }
    }
    for (int x = 1; x <= c; ++x) {
        if (dp[t1][t2][t3][a][b][c - x] == false) {
            dp[t1][t2][t3][a][b][c] = true;
            break;
        }
    }
}
}
}
}
}
}
}
}
}

vector<int> a = mx;

while (1) {
    auto p = find(a[0], a[1], a[2], used[0], used[1], used[2]);
    int i, x;
    i = p.first; x = p.second;
    cout << p.first << ' ' << p.second << endl;

    if (i == -1) {
        exit(0);
    }
    if (x == 0) {
        used[i - 1] = 0;
        a[i - 1] = mx[i - 1];
    } else {
        a[i - 1] -= x;
    }

    cin >> i >> x;
    if (i == -1) {
        exit(0);
    }
    if (x == 0) {
        used[i - 1] = 0;
        a[i - 1] = mx[i - 1];
    } else {
        a[i - 1] -= x;
    }
}
}
}

```


Task E ()

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>

using namespace std;

void encode() {
    int x; cin >> x;
    for (int i = 0; i < 10; ++i) {
        if ((x >> i) & 1) {
            for (int j = 0; j <= i; ++j)
                cout << 1;
            for (int j = i + 1; j < 10; ++j)
                cout << 0;
            cout << '\n';
        } else {
            for (int j = 0; j < 10; ++j)
                cout << 0;
            cout << '\n';
        }
    }
}

void decode() {
    int res = 0;
    for (int i = 0; i < 10; ++i) {
        int num = 0;
        for (int j = 0; j < 10; ++j) {
            char c; cin >> c;
            num += (c == '1');
        }
        if (num)
            res += (1 << (num - 1));
    }
    cout << res << '\n';
}

signed main() {
    int t; cin >> t;
    string s; cin >> s;
    while (t--) {
        if (s == "transmit")
            encode();
        else
            decode();
    }
}

/*
1111100000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000

1100000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
0000000000
*/
```


Task F ()

```
#include <iostream>
#include <string>
#include <algorithm>

using namespace std;

signed main() {
    string a, b;
    cin >> a >> b;
    string res;

    reverse(a.begin(), a.end());
    reverse(b.begin(), b.end());
    int n = max((int)a.size(), (int)b.size());
    while (a.size() < n)
        a += '0';
    while (b.size() < n)
        b += '0';
    while (res.size() < n)
        res += '0';
    for (int i = 0; i < n; ++i) {
        res[i] = (a[i] - '0' + b[i] - '0') + '0';
    }
    res += '0';
    for (int i = 0; i < n; ++i) {
        if (res[i] > '0' + 9) {
            res[i] = ((res[i] - '0') % 10) + '0';
            res[i + 1] += 1;
        }
    }

    while (res.back() == '0')
        res.pop_back();
    reverse(res.begin(), res.end());
    cout << res << '\n';
}
```