

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	70	10	28	65	373

Task A ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
using namespace std;
// #define int long long
#define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int, int>
vector<int> perm = { 0, 1, 2, 3, 4, 5 };
bool cmp(const int& a, const int& b) {
    return perm[a] < perm[b];
}

int32_t main()
{
    IOS;
    vector<int> d(6);
    for (int i = 0; i < 6; i++) {
        cin >> d[i];
        d[i]--;
    }
    vector<int> ans;
    while (next_permutation(all(perm))) {
        vector<int> yet;
        bool f = true;
        for (int i = 0; i < 6; i++) {
            yet.push_back(i);
            sort(all(yet), cmp);
            if (yet[d[i]] != i) {
                f = false;
            }
        }
        if (f) {
            ans = perm;
        }
    }
    if (ans.empty()) {
        ans = perm;
    }
    for (auto& c : ans) {
        cout << c + 1 << " ";
    }
}
```


Task B ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
using namespace std;
#define int long long
#define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int,int>
vector<int>perm = { 0,1,2,3,4,5 };
bool cmp(const int& a, const int& b) {
    return perm[a] < perm[b];
}
int32_t main()
{
    IOS;
    string s;
    cin >> s;
    const int dop = 1e5;
    const int mx = (10011 * 10011);
    if (s == "first") {
        int n;
        cin >> n;
        vector<int>a(n);
        int sum = 0;
        for (int i = 0; i < n; i++) {
            cin >> a[i];
            sum += a[i];
        }
        if (sum % n == 0) {
            cout << sum / n << "\n";
        }
        else {
            cout << sum * dop << "\n";
        }
    }
    else {
        int n;
        cin >> n;
        vector<int>b(n);
        int sum = 0;
        for (int i = 0; i < n; i++) {
            cin >> b[i];
            sum += b[i];
        }
        if (sum >= 1e5) {
            for (int j = 1; j <= mx; j++) {
                if (sum - (j * dop) * n < 1e5) {
                    cout << (sum - (j * dop * n)) + j << "\n";
                    return 0;
                }
            }
        }
        else {
            cout << sum << "\n";
        }
    }
}
```


Task C ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
#include <cassert>
#include <queue>
using namespace std;
#define int long long
#define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int, int>
vector<pii>ans;
int di[4] = { 1,0,-1,0 };
int dj[4] = { 0,1,0,-1 };
void fnd(vector<vector<int>>&state, int mx, int my) {
    vector<vector<int>>vis = state;
    for (int i = 1; i <= my; i++) {
        for (int j = 1; j <= mx; j++) {
            if (vis[i][j] != 0) continue;
            queue<pii>q;
            q.push({ i, j });
            int mxi = i;
            int mxj = j;
            int mini = i;
            int minj = j;
            while (!q.empty()) {
                pii cur = q.front();
                mxi = max(mxi, cur.first);
                mxj = max(mxj, cur.second);
                mini = min(mini, cur.first);
                minj = min(minj, cur.second);
                q.pop();
                if (vis[cur.first][cur.second] == 2) continue;
                vis[cur.first][cur.second] = 2;
                for (int g = 0; g < 4; g++) {
                    if (vis[cur.first + di[g]][cur.second + dj[g]] == 0) {
                        q.push({ cur.first + di[g], cur.second + dj[g] });
                    }
                }
            }
            bool f = true;
            for (int ci = mini; ci <= mxi; ci++) {
                for (int cj = minj; cj <= mxj; cj++) {
                    if (vis[ci][cj] != 2) {
                        f = false;
                    }
                }
            }
            if (f) {
                int a = mxi - mini + 1;
                int b = mxj - minj + 1;
                if (a > b) swap(a, b);
                ans.push_back({ a, b });
            }
        }
    }
}

bool add(int posi, int posj, vector<vector<int>>&state, int mi, int mj) {
    bool f = true;
    bool up = 0;
    bool lf = 0;
    //deb(state);
    vector<vector<int>>cstate = state;
    for (int i = 0; i < mi; i++) {
        for (int j = 0; j < mj; j++) {
```



```

        if (posi + i < 11 && posj + j < 11) {
            state[posi + i][posj + j]++;
            {
                if (posi + i - 1 >= 0) {
                    if (cstate[posi + i - 1][posj + j] == 1) {
                        up = true;
                    }
                }
                if (posi + i + 1 < 11) {
                    if (cstate[posi + i + 1][posj + j] == 1) {
                        up = true;
                    }
                }
            }
            {
                if (posj + j - 1 >= 0) {
                    if (cstate[posi + i][posj + j - 1] == 1) {
                        lf = true;
                    }
                }
                if (posj + j + 1 < 11) {
                    if (cstate[posi + i][posj + j + 1] == 1) {
                        lf = true;
                    }
                }
            }
            if (state[posi + i][posj + j] != 1) {
                f = false;
            }
        }
        else {
            f = false;
        }
    }
    return f && up && lf;
}

void dec(int posi, int posj, vector<vector<int>>& state, int mi, int mj) {
    bool f = true;
    bool up = 0;
    bool lf = 0;
    for (int i = 0; i < mi; i++) {
        for (int j = 0; j < mj; j++) {
            if (posi + i < 11 && posj + j < 11) {
                state[posi + i][posj + j]--;
                if (state[posi + i][posj + j] != 1) {
                    f = false;
                }
            }
            else {
                f = false;
            }
        }
    }
    return;
}

void rec(int can, vector<vector<int>>&state, int mx, int my, vector<pii>&box) {
    fnd(state, mx, my);
    for (int i = 1; i <= my; i++) {
        for (int j = 1; j <= mx; j++) {
            for (int g=0; g < box.size(); g++){
                if (!(can & (1ll << g))) continue;
                {
                    bool f = add(i, j, state, box[g].first, box[g].second);
                    if (f) {
                        rec(can - (1ll<<g), state, mx, my, box);
                    }
                    dec(i, j, state, box[g].first, box[g].second);
                }
            }
            {
                bool f = add(i, j, state, box[g].second, box[g].first);
                if (f) {
                    rec(can - (1ll << g), state, mx, my, box);
                }
            }
        }
    }
}

```


Task D ()

```

#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
#include <cassert>
#include <cmath>
using namespace std;
// #define int long long
#define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int, int>
int osn = 51;
map < pair<int, int>, pii> ans; // , /
vector<int> beg;
vector<int> clr;
pii lose = { -1, -1 };
void rec(int alive, vector<int> & state, int num) {
    if (ans.count({ alive, num })) {
        return;
    }
    if (alive == 0) {
        bool f = true;
        for (int i = 0; i < state.size(); i++) {
            if (state[i] != 0) f = false;
        }
        if (f) {
            ans[{ alive, num }] = { -1, -1 };
            return;
        }
    }
    bool f = false;
    for (int i = 0; i < state.size(); i++) {
        if (alive & (1ll << i)) {
            int was = state[i];
            state[i] = beg[i];
            num -= pow(osn, i) * was - pow(osn, i) * state[i];
            rec(alive ^ (1ll << i), state, num);
            if (!f && ans[{ alive ^ (1ll << i), num }] == lose) {
                num -= pow(osn, i) * state[i] - pow(osn, i) * was;
                state[i] = was;
                ans[{ alive, num }] = { i + 1, 0 };
                f = true;
                return;
            }
        }
        else {
            num -= pow(osn, i) * state[i] - pow(osn, i) * was;
            state[i] = was;
        }
    }
    for (int i = 0; i < state.size(); i++) {
        for (int j = 1; j <= state[i]; j++) {
            int val = j;
            if (state[i] - val >= 0) {
                state[i] -= val;
                num -= pow(osn, i) * val;
                rec(alive, state, num);
                pair<int, int> curch = { alive, num };
                if (!f && ans[curch] == lose) {
                    state[i] += val;
                    num += pow(osn, i) * val;
                    ans[{ alive, num }] = { i + 1, val };
                    f = true;
                    return;
                }
            }
        }
    }
}

```



```

        else {
            state[i] += val;
            num += pow(osn, i) * val;
        }
    }
}
if (!f) {
    ans[{alive, num}] = { -1, -1 };
}
return;
}
void ob_mv(int& alive, vector<int>& a, pii mv, int& num) {
    if (mv.second == 0) {
        num -= pow(osn, mv.first - 1) * a[mv.first - 1] - pow(osn, mv.first - 1) * beg[mv.first - 1];
        first - 1;
        a[mv.first - 1] = beg[mv.first - 1];
        alive ^= (1ll << (mv.first - 1));
    }
    else {
        a[mv.first - 1] -= mv.second;
        num -= pow(osn, mv.first - 1) * mv.second;
    }
}
int32_t main()
{
    IOS;
    int n;
    cin >> n;
    vector<int> a(n);
    int inn = 0;
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        inn += a[i] * pow(osn, i);
        clr.push_back(0);
    }
    int alive = (1ll << n) - 1;
    beg = a;
    if (pow(osn, 0) * 50 + pow(osn, 1) * 50 + pow(osn, 2) * 50 == inn) {
        assert(false);
    }
    vector<int> state = a;
    rec(alive, state, inn);
    //cout << rec(alive, state) << "\n";
    //pair<int, vector<int>>> ch = { 0, a };
    //cout << ans[ch].first << "\n";
    {
        int alive = (1ll << n) - 1;
        vector<int> state = a;
        int num = inn;
        while (1) {
            if (!ans.count({ alive, num })) {
                assert(false);
            }
            pii mv = ans[{alive, num}];
            if (mv.first == -1) {
                cout << mv.first << " " << mv.second << endl;
                cout.flush();
                return 0;
            }
            else {
                cout << mv.first << " " << mv.second << endl;
                cout.flush();
                ob_mv(alive, state, mv, num);
            }
            pii res;
            cin >> res.first >> res.second;
            if (res.first == -1) {
                return 0;
            }
            else {
                ob_mv(alive, state, res, num);
            }
        }
    }
}

```


} } }

Task E ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
#include <cassert>
#include <cmath>
using namespace std;
// #define int long long
// #define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int, int>
int32_t main()
{
    IOS;
    int t;
    cin >> t;
    string s;
    cin >> s;
    if (s == "transmit") {
        while (t--) {
            int n;
            cin >> n;
            for (int i = 0; i < 10; i++) {
                for (int j = 0; j < 10; j++) {
                    if (n) {
                        cout << 1;
                        n--;
                    }
                    else {
                        cout << 0;
                    }
                }
                cout << "\n";
            }
            cout << "\n";
        }
    }
    else {
        while (t--) {
            int ans = 0;
            for (int i = 0; i < 10; i++) {
                for (int j = 0; j < 10; j++) {
                    char c;
                    cin >> c;
                    ans += c - '0';
                }
            }
            cout << ans << "\n";
        }
    }
}
```


Task F ()

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
#include <set>
#include <string>
#include <vector>
#include <map>
#include <iomanip>
#include <algorithm>
#include <cassert>
#include <cmath>
using namespace std;
// #define int long long
#define double long double
#define IOS ios::sync_with_stdio(0); cin.tie(0); cout.tie(0);
#define all(x) x.begin(), x.end()
#define pii pair<int, int>
int checki(int pos, string& tek, string& res) {
    int cnt = 0;
    int len = tek.size();
    int cur = 0;
    for (int i = pos; i < res.size(); i++) {
        if (res[i] == tek[cur]) {
            cur++;
        }
        else {
            break;
        }
        if (cur == len) {
            cnt++;
            cur %= len;
        }
    }
    if (cnt > 1) return cnt;
    else return -1;
}
string compres(string& res) {
    int l = 0;
    string ans = "";
    while (l < res.size()) {
        pair<int, string> fndp = { -1, "" };
        string tek = "";
        for (int i = l; i < res.size(); i++) {
            tek += res[i];
            int val = checki(l, tek, res);
            if (val == -1) {
                continue;
            }
            else {
                fndp = { val, tek };
            }
        }
        if (fndp.first == -1) {
            ans += res[l];
            l++;
        }
        else {
            ans += "(" + fndp.second + "|" + to_string(fndp.first) + ")";
            l += (fndp.second.size()) * fndp.first;
        }
    }
    return ans;
}
int32_t main()
{
    IOS;
    string a;
    cin >> a;
    string b;
    cin >> b;
    while (a.size() < b.size()) {
        a = "0" + a;
    }
}
```



```

while (b.size() < a.size()) {
    b = "0" + b;
}
a = "0" + a;
b = "0" + b;
vector<int>res(a.size(), 0);
int mv = 0;
for (int i = b.size() - 1; i > -1; i--) {
    int sum = (a[i] + b[i] - 2.0 * '0') + mv;
    res[i] = sum % 10;
    mv = sum / 10;
}
reverse(all(res));
if (res.back() == 0) res.pop_back();
reverse(all(res));
{
    string in = "";
    for (int i = 0; i < res.size(); i++) {
        in += to_string(res[i]);
    }
    string ans = compres(in);
    cout << ans << "\n";
}
}

```