

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	52	10	462

Task A ()

```
#include "bits/stdc++.h"

using namespace std;

#define all(a) a.begin(), a.end()
#define vc vector
#define forn(i, n) for(int i = 0; i < (int)n ; i++)

void run(){
    vc<int> A(6);
    forn(i , 6) cin >> A[i];
    vc<int> p(6);
    forn(i , 6) p[i] = i + 1;

    do{
        vc<int> simul , b;

        forn(i , 6){
            int cnt = 0;
            for(int x : simul) {
                if(p[i] > x) cnt++;
            }
            b.push_back(cnt + 1);
            simul.push_back(p[i]);
        }
        if(b == A){
            for(int x : p) cout << x << " ";
            cout << "\n";
            return;
        }

    }while(next_permutation(all(p)));

}

signed main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    run();
    return 0;
}
```

Task B ()

```
#include "bits/stdc++.h"

using namespace std;

#define int long long
#define all(a) a.begin(), a.end()
#define vc vector
#define forn(i, n) for(int i = 0; i < (int)n ; i++)

#ifdef HOME
    #define debug(x) cerr << #x << ":_ " << x << endl;
#else
    #define debug(x) 11;
#endif

void code1(){
    int n;
    int x = 0;
    cin >> n;
    int s = 0;
    forn(i, n){
        int a;
        cin >> a;
        x += a;
    }
    s = x;
    x <<= 30;
    cout << x << endl;
}

void decode1(){
    int n;
    cin >> n;
    int sum = 0;
    forn(i, n){
        int a;
        cin >> a;
        if(i == 0){
            sum = (a >> 30);
        }
        debug(sum);
        int t = (a & ((1ll << (29)) - 1));
        debug(t);
        sum += t;
    }
    cout << sum << "\n";
}

void run(){
    string t;
    cin >> t;
    if(t == "first") code1();
    else decode1();
}

signed main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    run();
    return 0;
}
```

Task C ()

```
#include "bits/stdc++.h"

using namespace std;

// #define int long long
#define all(a) a.begin(), a.end()
#define vc vector
#define forn(i, n) for(int i = 0; i < (int)n ; i++)
#define pb push_back
#ifdef HOME
    #define debug(x) cerr << #x << ":_ " << x << endl;
#else
    #define debug(x) 11;
#endif

// vc<array<int , 2>>> ans;

inline bool cmax(int& a , int b){
    if(a < b){
        a = b;
        return 1;
    }
    return 0;
}

inline bool cmin(int& a , int b){
    if(a > b){
        a = b;
        return 1;
    }
    return 0;
}

char vis[13][13];
vc<array<int , 2>>> ans;
int mny , mnx , mxm , mxy , cnt;

void dfs(int x , int y , vc<vc<int>>> &pole){
    if(pole[x][y]) return;
    pole[x][y] = 1;
    cmax(mxy , y);
    cmax(mxm , x);
    cmin(mny , y);
    cmin(mnx , x);
    cnt++;
    dfs(x + 1 , y , pole);
    dfs(x - 1 , y , pole);
    dfs(x , y - 1 , pole);
    dfs(x , y + 1 , pole);
}

void add(vc<vc<int>>> pole){
    // for(auto x : pole) {
    //     for(int y : x) cout << y;
    //     cout << endl;
    // }
    // cout << endl;

    for(int i = 0 ; i < pole.size(); i++){
        for(int j = 0 ; j < pole[i].size() ; j++){
            if(pole[i][j]) continue;
            mxm = mxy = -100;
            mny = mnx = 100;
            cnt = 0;
            dfs(i , j , pole);

            int a = (mxm - mnx + 1);
            int b = (mxy - mny + 1);
        }
    }
}
```

```

        if(cnt == a * b){
            // debug(a);
            // debug(b);
            ans.pb({min(a , b) , max(a , b)});
        }
    }
}

void run(){
    vc<array<int , 2>> A(3);
    forn(i , 3) cin >> A[i][0] >> A[i][1];

    vc<int> p = { 0 , 1 , 2};

    do{
        vc<vc<int>> pole(A[p[0]][0] + 2 , vc<int>(A[p[0]][1] + 2 , 1));
        forn(j , A[p[0]][0]){
            forn(z , A[p[0]][1]){
                pole[j + 1][z + 1] = 0;
            }
        }
        auto check = [&](int x , int y , int szx , int szy){
            for(int i = x ; i < x + szx ; i++){
                for(int j = y ; j < y + szy ; j++){
                    if(pole[i][j]) return false;
                }
            }
            bool f1 = 0 , f2 = 0;
            for(int i = x ; i < x + szx ; i++){
                for(int j = y ; j < y + szy ; j++){
                    f1 |= pole[i][j + 1];
                    f1 |= pole[i][j - 1];
                    f2 |= pole[i + 1][j];
                    f2 |= pole[i - 1][j];
                }
            }
            return f1 && f2;
        };

        auto fill = [&](int x , int y , int szx , int szy , int g){
            for(int i = x ; i < x + szx ; i++){
                for(int j = y ; j < y + szy ; j++){
                    assert(pole[i][j] != g);
                    pole[i][j] = g;
                }
            }
        };

        add(pole);
        for(int mask = 0 ; mask < 4 ; mask++){
            if(mask & 1) swap(A[p[1]][0] , A[p[1]][1]);
            if(mask & 2) swap(A[p[2]][0] , A[p[2]][1]);

            for(int i = 1 ; i <= A[p[0]][0] ; i++){
                for(int j = 1 ; j <= A[p[0]][1] ; j++){
                    if(check(i , j , A[p[1]][0] , A[p[1]][1])){
                        fill(i , j , A[p[1]][0] , A[p[1]][1] , 1);
                        add(pole);

                        for(int i1 = 1 ; i1 <= A[p[0]][0] ; i1++){
                            for(int j1 = 1 ; j1 <= A[p[0]][1] ; j1++){
                                if(check(i1 , j1 , A[p[2]][0] , A[p[2]][1])){
                                    fill(i1 , j1 , A[p[2]][0] , A[p[2]][1] , 1);
                                    add(pole);
                                    fill(i1 , j1 , A[p[2]][0] , A[p[2]][1] , 0);
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

        }
        fill(i, j, A[p[1]][0], A[p[1]][1], 0);
    }
}

if(mask & 1) swap(A[p[1]][0], A[p[1]][1]);
if(mask & 2) swap(A[p[2]][0], A[p[2]][1]);

}

} while(next_permutation(all(p)));
sort(all(ans));
ans.resize(unique(all(ans)) - ans.begin());
for(auto x : ans) cout << x[0] << " " << x[1] << "\n";
}

signed main() {
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    run();
    return 0;
}

```

Task D ()

```
#include "bits/stdc++.h"

using namespace std;

// #define int long long
#define all(a) a.begin(), a.end()
#define vc vector
#define forn(i, n) for(int i = 0; i < (int)n ; i++)
#define pb push_back
#ifdef HOME
    #define debug(x) cerr << #x << ":_ " << x << endl;
#else
    #define debug(x) 11;
#endif

// vc<array<int , 2>> ans;

inline bool cmax(int& a , int b){
    if(a < b){
        a = b;
        return 1;
    }
    return 0;
}

inline bool cmin(int& a , int b){
    if(a > b){
        a = b;
        return 1;
    }
    return 0;
}

int n;
int calc(vc<int> A , vc<int> B){
    int X = 0;
    forn(i , n){
        if(B[i] == 0) {
            X ^= A[i];
            continue;
        }
        if(B[i] == A[i]) X ^= (A[i] + 1);
        else X ^= A[i];
    }
    return X;
}

void run(){
    // int n;
    cin >> n;
    vc<int> A(n);
    vc<int> B(n);
    forn(i , n){

        cin >> A[i];
        B[i] = A[i];
    }
    if(calc(A , B) == 0) {
        cout << "-1_-1" << endl;
        return;
    }

    while(1){
        int a = - 2 , b = 0;
        // cout << "game :";
        // cout << A[0] << " " << B[0] << endl;
        forn(i , n){
            if(B[i]) {
                int mema = A[i];
```

```

        int memb = B[i];
        A[i] = B[i];
        B[i] = 0;
        if (calc(A, B) == 0){

            A[i] = mema;
            B[i] = memb;

            a = i;
            b = 0;
            break;
        }
        A[i] = mema;
        B[i] = memb;
    }
    for (int j = 1; j <= A[i]; j++){
        A[i] -= j;
        if (calc(A, B) == 0){
            A[i] += j;
            a = i;
            b = j;
            break;
        }
        A[i] += j;
    }

}

assert(a != -2);
cout << a + 1 << "\n" << b << endl;
if (b == 0) {
    A[a] = B[a];
    B[a] = 0;
}
else {
    A[a] -= b;
}

int x, y;
cin >> x >> y;
if (x == -1 && y == -1) return;
if (y == 0) {
    A[x - 1] = B[x - 1];
    B[x - 1] = 0;
}
else {
    A[x - 1] -= y;
}
}

}

signed main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    run();
    return 0;
}

```

Task E ()

```
#include "bits/stdc++.h"

using namespace std;

// #define int long long
#define all(a) a.begin(), a.end()
#define vc vector
#define forn(i, n) for(int i = 0; i < (int)n ; i++)
#define pb push_back

#ifdef HOME
    #define debug(x) cerr << #x << ":_" << x << endl;
#else
    #define debug(x) 11;
#endif

int cnt = 0;
bool ready = 0;

const int maxn = 1e6;
#define int signed char

pair<vc<int> , vc<int>>> A[maxn];
map<pair<vc<int> , vc<int>>> , long long> AA;

int B[10];

void rec(int ls , vc<int> &a , vc<int> &b){
    forn(i , 10) B[i] = b[i];
    for(int x : a){
        sort(B , B + 10);
        reverse(B , B + 10);
        for(int i = 0 ; i < 10 ; i++){
            if(x && B[i]){
                x--;
                B[i]--;
            }
        }
        if(x) return;
    }
    if(a.size() == 10){
        forn(i , 10) if(B[i]) return;
        A[++cnt] = {a , b};
        AA[{a , b}] = cnt;
        return;
    }
    for(int i = ls; i <= 10 ; i++){
        a.pb(i);
        rec(i , a , b);
        a.pop_back();
    }
}

void R(int ls , vc<int> &a){
    if(ready) return;
    if(a.size() == 10){
        // cnt = 0;
        // for(int x : a) cout << x << " ";
        // cout << endl;
        vc<int> t;
        rec(0 , t , a);
        // debug(cnt);
        if(cnt > 1e5) {
            ready = 1;
        }
        // cnt++;
        return;
    }
}
```



```

    }
    for(int i = 1s; i <= 10 ; i++){
        a.pb(i);
        R(i , a );
        a.pop_back();
    }
}

void code1(){
    long long n;
    cin >> n;

    vc<int> a = A[n].first;
    vc<int> b = A[n].second;
    vc<vc<int>> pole(10 , vc<int>(10 , '0'));

    // for(int x : a) cout << x << " ";
    // cout << endl;
    // for(int x : b) cout << x << " ";
    // cout << endl;

    forn(i , 10){
        vc<pair<int , int>> aa(10);
        forn(j , 10) aa[j] = {a[j] , j};
        sort(all(aa));
        reverse(all(aa));
        forn(z , b[i]) {
            pole[i][aa[z].second] = '1';
            a[aa[z].second]--;
        }
    }
    forn(i , 10){
        forn(j , 10) cout << pole[i][j];
        cout << "\n";
    }
    // cout << endl;
}

pair<vc<int> , vc<int>> calc(vc<vc<int>> pole){
    vc<int> a , b;
    for(int i = 0 ; i < 10 ; i++){
        int s = 0;
        for(int j = 0 ; j < 10 ; j++) s += pole[i][j];
        a.pb(s);
    }
    for(int i = 0 ; i < 10 ; i++){
        int s = 0;
        for(int j = 0 ; j < 10 ; j++) s += pole[j][i];
        b.pb(s);
    }
    sort(all(a));
    sort(all(b));
    return {b, a};
}

void decode1(){
    vc<vc<int>> pole(10 , vc<int>(10));
    forn(i , 10){
        forn(j , 10){
            char a;
            cin >> a;
            pole[i][j] = a - '0';
        }
    }
    cout << AA[calc(pole)] << "\n";
}
}

```

```

void run(){
    vc<int> tt;
    R(4 , tt);
    debug("ok");
    long long n;
    cin >> n;
    string t;
    cin >> t;
    while(n--){

        if(t == "transmit") code1();
        else decode1();
    }
}

signed main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    run();
    return 0;
}

```

Task F ()

```
a = int(input())  
b = int(input())  
print(a + b)
```