

# Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	80	60	28	10	378

## Task A ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#include <ext/pb_ds/detail/standard_policies.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define endl '\n'
#define debug(x) cerr << (#x) << "==" << (x) << endl;

using ordered_set = tree<pair<int, int>, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update>;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

const int MAXN = 2e5 + 1;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int MOD2 = 1e9 + 9;
const int P = 97;
const ld Pi = 3.1415926536;
const ld E = 1e-10;
const ld rad = 360.0 / (2 * Pi);
const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(RANDOM);
struct chash {
    int operator()(int x) const { return x ^ RANDOM; }
};

void Fast(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
}

signed main(){
    Fast();
    vector<int> d(6);
    vector<int> k;
    for (int i = 0; i < 6; i++){
        cin >> d[i];
        vector<int> g;
        for (int j = 0; j < d[i] - 1; j++) g.push_back(k[j]);
        g.push_back(i);
        for (int j = d[i] - 1; j < k.size(); j++) g.push_back(k[j]);
        k = g;
    }
    vector<int> ans(6);
    for (int i = 0; i < 6; i++){
        ans[k[i]] = i + 1;
    }
    for (auto i:ans){
```



```
        cout << i << " ";  
    }  
    return 0;  
}
```



## Task B ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#include <ext/pb_ds/detail/standard_policies.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define endl '\n'
#define debug(x) cerr << (#x) << "==" << (x) << endl;

using ordered_set = tree<pair<int, int>, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update>;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

const int MAXN = 2e5 + 1;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int MOD2 = 1e9 + 9;
const int P = 97;
const ld Pi = 3.1415926536;
const ld E = 1e-10;
const ld rad = 360.0 / (2 * Pi);
const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(RANDOM);
struct chash {
    int operator()(int x) const { return x ^ RANDOM; }
};

void Fast(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
}

signed main(){
    Fast();
    string s;
    cin >> s;
    if (s == "first"){
        int n;
        cin >> n;
        int sum = 0;
        for (int i = 0; i < n; i++){
            int a;
            cin >> a;
            sum += a;
        }
        cout << sum << 100;
    } else {
        int n;
        cin >> n;
        int sum = 0;
        int g = 0;
        for (int i = 0; i < n; i++){
            int a;
            cin >> a;
            a %= 100;
            sum += (a % 1000);
            g = (a / 1000);
        }
        cout << sum + g;
    }
    return 0;
}
```



## Task C ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#include <ext/pb_ds/detail/standard_policies.hpp>

using namespace std;
using namespace __gnu_pbds;

// #define int long long
#define endl '\n'
#define debug(x) cerr << (#x) << " = " << (x) << endl;

using ordered_set = tree<pair<int, int>, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update>;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

const int MAXN = 2e5 + 1;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int MOD2 = 1e9 + 9;
const int P = 97;
const ld Pi = 3.1415926536;
const ld E = 1e-10;
const ld rad = 360.0 / (2 * Pi);
const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(RANDOM);
struct chash {
    int operator()(int x) const { return x ^ RANDOM; }
};

void Fast(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
}

set<pair<int, int>> ans;

void f(int a1, int b1, int a2, int b2, int a3, int b3){
    if (b1 > b2 && a1 >= a2){
        if (a1 == a2){
            ans.insert({min(b1 - b2, a1), max(b1 - b2, a1)});
            if (a3 == a1 && b1 > b2 + b3){
                ans.insert({min(b1 - b2 - b3, a1), max(b1 - b2 - b3, a1)});
                ans.insert({min(b1 - b3, a1), max(b1 - b3, a1)});
            }
            if (b3 == a1 && b1 > b2 + a3){
                ans.insert({min(b1 - b2 - a3, a1), max(b1 - b2 - a3, a1)});
                ans.insert({min(b1 - a3, a1), max(b1 - a3, a1)});
            }
            if (b1 - b2 == a3 && b3 < a1){
                ans.insert({min(a1 - b3, a3), max(a1 - b3, a3)});
            }
            if (b1 - b2 == b3 && a3 < a1){
                ans.insert({min(a1 - a3, b3), max(a1 - a3, b3)});
            }
        }
        if (a2 == a3){
            if (b2 + b3 == b1){
                ans.insert({min(b1, a2), max(b1, a2)});
            }
        }
        if (a2 == b3){
            if (b2 + a3 == b1){
                ans.insert({min(b1, a2), max(b1, a2)});
            }
        }
        if (a1 - a2 == a3){
            if (b2 == b3){

```



```

        ans.insert({min(b1 - b2, a1), max(b1 - b2, a1)});
    }
    if (b2 + b3 == b1){
        ans.insert({min(b2, a3), max(b2, a3)});
        ans.insert({min(b3, a2), max(b3, a2)});
    }
}
if (a1 - a2 == b3){
    if (b2 == a3){
        ans.insert({min(b1 - b2, a1), max(b1 - b2, a1)});
    }
    if (b2 + a3 == b1){
        ans.insert({min(b2, b3), max(b2, b3)});
        ans.insert({min(a3, a2), max(a3, a2)});
    }
}
if (a1 - a2 < a3 && b1 >= b2 + b3 && a1 > a2 && a1 >= a3){
    ans.insert({min(a1 - a2, b2), max(a1 - a2, b2)});
}
if (a1 - a2 < b3 && b1 >= b2 + a3 && a1 > a2 && a1 >= b3){
    ans.insert({min(a1 - a2, b2), max(a1 - a2, b2)});
}
if (b1 - b2 < a3 && a1 >= a2 + b3 && b1 >= a3){
    ans.insert({min(b1 - b2, a2), max(b1 - b2, a2)});
}
if (b1 - b2 < b3 && a1 >= a2 + a3 && b1 >= b3){
    ans.insert({min(b1 - b2, a2), max(b1 - b2, a2)});
}
}
if (b1 == b2 && a1 > a2){
    ans.insert({min(a1 - a2, b1), max(a1 - a2, b1)});
    if (a1 == a3 + a2 && b3 < b1){
        ans.insert({min(a3, b1 - b3), max(a1 - a2, b1 - b3)});
    }
    if (a1 == b3 + a2 && a3 < b1){
        ans.insert({min(b3, b1 - a3), max(b3, b1 - a3)});
    }
    if (b3 == b1 && a1 > a2 + a3){
        ans.insert({min(a1 - a2 - a3, b1), max(a1 - a2 - a3, b1)});
    }
    if (a3 == b1 && a1 > a2 + b3){
        ans.insert({min(a1 - a2 - b3, b1), max(a1 - a2 - b3, b1)});
    }
}
if (b1 == b3 && a1 > a3){
    ans.insert({min(b1, a1 - a3), max(b1, a1 - a3)});
    if (a1 - a3 == a2 && b1 > b2){
        ans.insert({min(a1 - a3, b1 - b2), max(a1 - a3, b1 - b2)});
    }
    if (a1 - a3 == b2 && b1 > a2){
        ans.insert({min(a1 - a3, b1 - a2), max(a1 - a3, b1 - a2)});
    }
}
if (b1 == a3 && a1 > b3){
    ans.insert({min(b1, a1 - b3), max(b1, a1 - b3)});
    if (a1 - b3 == a2 && b1 > b2){
        ans.insert({min(a1 - b3, b1 - b2), max(a1 - b3, b1 - b2)});
    }
    if (a1 - b3 == b2 && b1 > a2){
        ans.insert({min(a1 - b3, b1 - a2), max(a1 - b3, b1 - a2)});
    }
}
}

if (a3 == a1){
    if (b1 > b3){
        ans.insert({min(b1 - b3, a1), max(b1 - b3, a1)});
    }
    if (b1 - b3 == a2 && a1 > b2){
        ans.insert({min(a1 - b2, a2), max(a1 - b2, a2)});
    }
}
if (b1 - b3 == b2 && a1 > a2){

```



```

        ans.insert({min(a1 - a2, b2), max(a1 - a2, b2)});
    }
}
if (b3 == a1){
    if (b1 > a3){
        ans.insert({min(b1 - a3, a1), max(b1 - a3, a1)});
    }
    if (b1 - a3 == a2 && a1 > b2){
        ans.insert({min(a1 - b2, a2), max(a1 - b2, a2)});
    }
    if (b1 - a3 == b2 && a1 > a2){
        ans.insert({min(a1 - a2, b2), max(a1 - a2, b2)});
    }
}
}

signed main(){
    Fast();
    int a1, b1, a2, b2, a3, b3;
    cin >> a1 >> b1 >> a2 >> b2 >> a3 >> b3;
    vector <pair <int, int>> p = {{max(a1, b1), min(a1, b1)}, {max(a2, b2), min(a2, b2)}, {max(a3,
        b3), min(a3, b3)}};
    sort(p.begin(), p.end());
    reverse(p.begin(), p.end());
    a1 = p[0].first, b1 = p[0].second;
    a2 = p[1].first, b2 = p[1].second;
    a3 = p[2].first, b3 = p[2].second;
    ans = {{min(a1, b1), max(a1, b1)}, {min(a2, b2), max(a2, b2)}, {min(a3, b3), max(a3, b3)}};
    f(a1, b1, a2, b2, a3, b3);
    f(a1, b1, a3, b3, a2, b2);
    f(a1, b1, b2, a2, a3, b3);
    f(a1, b1, b3, a3, a2, b2);
    f(a2, b2, a3, b3, a1, b1);
    f(a2, b2, b3, a3, a1, b1);
    for (auto i:ans){
        cout << i.first << " " << i.second << endl;
    }
    return 0;
}

```



## Task D ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#include <ext/pb_ds/detail/standard_policies.hpp>

using namespace std;
using namespace __gnu_pbds;

#define int long long
#define endl '\n'
#define debug(x) cerr << (#x) << " = " << (x) << endl;

using ordered_set = tree<pair<int, int>, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update>;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

const int MAXN = 2e5 + 1;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int MOD2 = 1e9 + 9;
const int P = 97;
const ld Pi = 3.1415926536;
const ld E = 1e-10;
const ld rad = 360.0 / (2 * Pi);
const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(RANDOM);
struct chash {
    int operator()(int x) const { return x ^ RANDOM; }
};

void Fast(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
}

signed main(){
    Fast();
    int n;
    cin >> n;
    vector<int> a(n);
    vector<int> c(n);
    vector<bool> used(n, true);
    int x = 0;
    for (int i = 0; i < n; i++) cin >> a[i], x ^= (a[i] + 1), c[i] = a[i] + 1;
    vector<int> b = a;
    if (x == 0){
        cout << -1 << " " << -1 << endl;
        cout.flush();
    }else{
        int k = 0;
        while (k != -1){
            for (int i = 0; i < n; i++){
                bool flag = false;
                for (int j = 0; j <= a[i]; j++){
                    if (j == 0 && used[i]){
                        if (((x ^ c[i]) ^ b[i]) == 0){
                            flag = true;
                            used[i] = false;
                            x = ((x ^ c[i]) ^ b[i]);
                            a[i] = b[i];
                            c[i] = b[i];
                            cout << i + 1 << " " << 0 << endl;
                            cout.flush();
                            break;
                        }
                    }
                }
            }
            if (flag == false){
                if (((x ^ c[i]) ^ (a[i] - j)) == 0){
                    flag = true;
                }
            }
        }
    }
}
```



```

        x = ((x ^ c[i]) ^ (a[i] - j));
        a[i] -= j;
        c[i] = a[i];
        cout << i + 1 << "└" << j << endl;
        cout.flush();
        break;
    }
}
}
if (flag) break;
}
int g;
cin >> k >> g;
if (k == -1) break;
k--;
if (g == 0){
    x = ((x ^ c[k]) ^ b[k]);
    a[k] = b[k];
    c[k] = b[k];
    used[k] = false;
} else {
    x = ((x ^ c[k]) ^ (a[k] - g));
    a[k] -= g;
    c[k] = a[k];
}
}
}
return 0;
}

```



## Task E ()

```
#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
#include <ext/pb_ds/detail/standard_policies.hpp>

using namespace std;
using namespace __gnu_pbds;

// #define int long long
#define endl '\n'
#define debug(x) cerr << (#x) << " = " << (x) << endl;

using ordered_set = tree<pair<int, int>, null_type, less<>, rb_tree_tag,
    tree_order_statistics_node_update>;

typedef long long ll;
typedef long double ld;
typedef unsigned long long ull;

const int MAXN = 2e5 + 1;
const int INF = 1e18;
const int MOD = 1e9 + 7;
const int MOD2 = 1e9 + 9;
const int P = 97;
const ld Pi = 3.1415926536;
const ld E = 1e-10;
const ld rad = 360.0 / (2 * Pi);
const int RANDOM = chrono::high_resolution_clock::now().time_since_epoch().count();
mt19937 rnd(RANDOM);
struct chash {
    int operator()(int x) const { return x ^ RANDOM; }
};

void Fast(){
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
}

signed main(){
    Fast();
    int t;
    cin >> t;
    vector<vector<short>> g = {{}};
    for (int i = 0; i < 10; i++){
        vector<vector<short>> k;
        for (auto j:g){
            vector<short> a = j;
            if (a.size() == 0){
                for (int y = 0; y < 11; y++){
                    a.push_back(y);
                    k.push_back(a);
                    a.pop_back();
                }
            } else {
                for (int y = a.back(); y < 11; y++){
                    a.push_back(y);
                    k.push_back(a);
                    a.pop_back();
                }
            }
        }
        g = k;
    }
    sort(g.begin(), g.end());
    string s;
    cin >> s;
    if (s[0] == 't'){
        while (t--){
            int n;
            cin >> n;
            vector<short> h = g[n - 1];
        }
    }
}
```



```

        for (int i = 0; i < 10; i++){
            for (int j = 0; j < h[i]; j++){
                cout << 1;
            }
            for (int j = 0; j < 10 - h[i]; j++){
                cout << 0;
            }
            cout << endl;
        }
        cout << endl;
    }
} else {
    while (t--){
        vector <short> h;
        for (int i = 0; i < 10; i++){
            string s;
            cin >> s;
            short sum = 0;
            for (int j = 0; j < 10; j++){
                sum += (s[j] - '0');
            }
            h.push_back(sum);
        }
        sort(h.begin(), h.end());
        cout << lower_bound(g.begin(), g.end(), h) - g.begin() + 1 << endl;
    }
}
return 0;
}

```



## Task F ()

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