

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	32	65	497

Task A ()

```
#include <bits/stdc++.h>
using namespace std;
#define int int64_t
#define double long double
// #define MULTITASK
#define NO_INTERACTIVE
void solve();
int32_t main() {
#ifdef BIBIKOV
    clock_t tStart = clock();
#endif
#ifdef NO_INTERACTIVE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
#ifdef NO_INTERACTIVE
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
#endif
    int tt = 1;
#ifdef MULTITASK
    cin >> tt;
#endif
    while (tt--) {
        solve();
        cout << '\n';
    }
#ifdef BIBIKOV
    fprintf(stderr, ">>_Runtime:_%{.10fs}\n", (double_t) (clock() - tStart) / CLOCKS_PER_SEC);
#endif
}
const int64_t INF = 1e18, inf = 2e9, mod = 998244353;

void solve() {
    int n = 6;
    vector<int> num(n);
    for (int i = 0, p; i < n; i++) {
        cin >> p;
        for (int j = 0; j < i; j++) {
            if (num[j] >= p)
                num[j]++;
        }
        num[i] = p;
    }
    for (int i = 0; i < n; i++)
        cout << num[i] << ' ';
}
```


Task B ()

```
#include <bits/stdc++.h>
using namespace std;
#define int int64_t
#define double long double
// #define MULTITASK
#define NO_INTERACTIVE
void solve1();
void solve2();
int32_t main() {
#ifdef BIBIKOV
    clock_t tStart = clock();
#endif
#ifdef NO_INTERACTIVE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
#ifdef NO_INTERACTIVE
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
#endif
    int tt = 1;
#ifdef MULTITASK
    cin >> tt;
#endif
    while (tt--) {
        string s;
        cin >> s;
        if (s == "first")
            solve1();
        else
            solve2();
        cout << '\n';
    }
#ifdef BIBIKOV
    fprintf(stderr, ">>Runtime: %.10fs\n", (double_t) (clock() - tStart) / CLOCKS_PER_SEC);
#endif
}
const int64_t INF = 1e18, inf = 2e9, mod = 998244353;

void solve1() {
    int n, s = 0;
    cin >> n;
    for (int i = 0, p; i < n && cin >> p; i++)
        s += p;
    cout << s * 100000;
}

void solve2() {
    int n, s = 0;
    cin >> n;
    for (int i = 0, p; i < n && cin >> p; i++)
        s += p;
    int p = s / 1e5 / n;
    s -= p * n * 1e5;
    s += p;
    cout << s;
}
```


Task C ()

```
#include <bits/stdc++.h>
using namespace std;
#define int int64_t
#define double long double
// #define MULTITASK
#define NO_INTERACTIVE
void solve();
int32_t main() {
#ifdef BIBIKOV
    clock_t tStart = clock();
#endif
#ifdef NO_INTERACTIVE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
#ifdef NO_INTERACTIVE
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
#endif
    int tt = 1;
#ifdef MULTITASK
    cin >> tt;
#endif
    while (tt--) {
        solve();
        cout << '\n';
    }
#ifdef BIBIKOV
    fprintf(stderr, ">>_Runtime: %.10fs\n", (double_t) (clock() - tStart) / CLOCKS_PER_SEC);
#endif
}
const int64_t INF = 1e18, inf = 2e9, mod = 998244353;

void solve() {
    vector<pair<int, int>> a(3);
    for (int i = 0; i < 3; i++) {
        cin >> a[i].first >> a[i].second;
        if (a[i].first < a[i].second)
            swap(a[i].first, a[i].second);
    }
    sort(a.rbegin(), a.rend());
    set<pair<int, int>> s;
    for (int i = 0; i < 3; i++)
        s.insert(a[i]);

    for (int i = 0; i < 8; i++) {
        int a1, b1, a2, b2, a3, b3;

        if (i & 1)
            a1 = a[0].first, b1 = a[0].second;
        else
            a1 = a[0].second, b1 = a[0].first;
        if (i & 2)
            a2 = a[1].first, b2 = a[1].second;
        else
            a2 = a[1].second, b2 = a[1].first;
        if (i & 4)
            a3 = a[2].first, b3 = a[2].second;
        else
            a3 = a[2].second, b3 = a[2].first;

        cout << a1 << ' ' << b1 << '\n' << a2 << ' ' << b2 << '\n' << a3 << ' ' << b3 << "\n\n";
        s.clear();

        if (a2 <= a1 && a3 <= a1 && a1 <= a2 + a3 && b2 + b3 == b1) {
```



```

        s.insert({max(a1 - a3, b1 - b2), min(a1 - a3, b1 - b2)});
        s.insert({max(a1 - a2, b1 - b3), min(a1 - a2, b1 - b3)});
    }

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

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

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

    if (a3 == a1 && a2 == a1 && b3 + b2 + b1 < 2 * max({b1, b2, b3})) {
        s.insert({max(a1, 2 * max({b1, b2, b3}) - (b3 + b2 + b1)), min(a1, 2 * max({b1, b2, b3}
            }) - (b3 + b2 + b1))});
    }

    //      set<pair<int, int>> ans;
    //      for (auto [b, a] : s)
    //          ans.insert({a, b});
    //      for (auto [a, b] : ans)
    //          cout << a << ' ' << b << '\n';
    //      cout << "\n\n";
    }
    set<pair<int, int>> ans;
    for (auto [b, a] : s)
        if (0 < a && 0 < b)
            ans.insert({a, b});
    for (auto [a, b] : ans)
        cout << a << '⌣' << b << '\n';
}

```


Task D ()

```
#include <bits/stdc++.h>
using namespace std;
#define int int64_t
#define double long double
// #define MULTITASK
// #define NO_INTERACTIVE
void solve();
int32_t main() {
#ifdef BIBIKOV
    clock_t tStart = clock();
#endif
#ifdef NO_INTERACTIVE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
#ifdef NO_INTERACTIVE
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
#endif
    int tt = 1;
#ifdef MULTITASK
    cin >> tt;
#endif
    while (tt--) {
        solve();
#ifdef NO_INTERACTIVE
        cout << '\n';
#endif
    }
#ifdef BIBIKOV
    fprintf(stderr, ">>_Runtime: %.10fs\n", (double_t) (clock() - tStart) / CLOCKS_PER_SEC);
#endif
}
const int64_t INF = 1e18, inf = 2e9, mod = 998244353;

void solve() {
    int n;
    cin >> n;
    vector<int> a(n), h(n);
    vector<bool> f(n, true);
    for (int i = 0; i < n; i++)
        cin >> a[i];
    h = a;
    int x = 0;
    for (int i = 0; i < n; i++)
        x ^= (a[i] + (a[i] == h[i] && f[i]));
    if (!x)
        return void(cout << -1 << '\n' << -1 << endl);
    int p = -1, v = -1;
    for (int i = 0; i < n; i++) {
        if (!(x ^ (a[i] + (a[i] == h[i] && f[i])) ^ h[i]) && f[i])
            p = i + 1, v = 0;
        for (int k = 1; k <= a[i]; k++) {
            if (!(x ^ (a[i] + (a[i] == h[i] && f[i])) ^ (a[i] - k)))
                p = i + 1, v = k;
        }
    }
    cout << p << '\n' << v << endl;
    if (p == -1)
        return;
    p--;
    x ^= (a[p] + (a[p] == h[p] && f[p]));
    if (v)
        a[p] -= v;
}
```



```

else
    a[p] = h[p], f[p] = false;
x ^= (a[p] + (a[p] == h[p] && f[p]));
while (cin >> p >> v) {
    if (p == -1 && v == -1)
        return;
    p--;
    x ^= (a[p] + (a[p] == h[p] && f[p]));
    if (v)
        a[p] -= v;
    else
        a[p] = h[p], f[p] = false;
    x ^= (a[p] + (a[p] == h[p] && f[p]));
    p = -1, v = -1;
    for (int i = 0; i < n; i++) {
        if (!(x ^ (a[i] + (a[i] == h[i] && f[i])) ^ h[i]) && f[i])
            p = i + 1, v = 0;
        for (int k = 1; k <= a[i]; k++) {
            if (!(x ^ (a[i] + (a[i] == h[i] && f[i])) ^ (a[i] - k)))
                p = i + 1, v = k;
        }
    }
    cout << p << ' ' << v << endl;
    if (p == -1)
        return;
    p--;
    x ^= (a[p] + (a[p] == h[p] && f[p]));
    if (v)
        a[p] -= v;
    else
        a[p] = h[p], f[p] = false;
    x ^= (a[p] + (a[p] == h[p] && f[p]));
}
}

```


Task E ()

```
#include <bits/stdc++.h>
using namespace std;
#define int int64_t
#define double long double
#define MULTITASK
#define NO_INTERACTIVE
void solve1();
void solve2();
int32_t main() {
#ifdef BIBIKOV
    clock_t tStart = clock();
#endif
#ifdef NO_INTERACTIVE
    freopen("input.txt", "r", stdin);
    freopen("output.txt", "w", stdout);
#endif
#ifdef NO_INTERACTIVE
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);
    cout.tie(nullptr);
#endif
    int tt = 1;
#ifdef MULTITASK
    cin >> tt;
#endif
    string s;
    cin >> s;
    while (tt--) {
        if (s == "transmit")
            solve1();
        else
            solve2();
        cout << '\n';
    }
#ifdef BIBIKOV
    fprintf(stderr, ">>Runtime: %.10fs\n", (double_t) (clock() - tStart) / CLOCKS_PER_SEC);
#endif
}
const int64_t INF = 1e18, inf = 2e9, mod = 998244353;
```

```
void solve1() {
    int n;
    cin >> n;
    vector<vector<int>> ans(10, vector<int>(10));
    for (int p = 1, i = 0, h = 1; i < 10 && p <= n; p *= 2, h++) {
        if (n & p) {
            for (int j = 0; j < min(10ll, h); j++)
                ans[i][j] = 1;
            i++;
        }
    }
    for (int i = 0; i < 10; i++) {
        for (int j = 0; j < 10; j++)
            cout << ans[i][j];
        cout << '\n';
    }
}
```

```
void solve2() {
    vector<vector<int>> a(10, vector<int>(10));
    for (int i = 0; i < 10; i++) {
        string s;
        cin >> s;
        for (int j = 0; j < 10; j++)
            a[i][j] = s[j] - '0';
    }
}
```



```

int ans = 0;
for (int i = 0, t = 0; i < 10; i++) {
    int p = accumulate(a[i].begin(), a[i].end(), 0ll) - 1;
    if (0 <= p && p < 9)
        ans += (1 << p);
    else if (0 <= p)
        ans += (1 << (p + t)), t++;
}
cout << ans << '\n';
}

```


Task F ()

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