

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	50	10	52	65	377

Task A ()

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

using namespace std;

signed main()
{
    vector<int> z(6), num(6);
    int c = 0;
    for (auto &i : z) cin >> i;
    for (int i = 1; i <= 6; i++) {
        c = 0;
        for (int j = 5; j >= 0; j--) {
            if (c==0 && z[j] == i) {
                num[j] = i;
                c++;
            }
            else if (c == 1) {
                z[j]++;
            }
        }
    }
    for (auto &i : num) cout << i << ' ';
}
```

Task B ()

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

using namespace std;

signed main()
{
    string s;
    cin >> s;
    long long n;
    cin >> n;
    if (s[0] == 'f') {
        long long sum = 0;
        for (long long i = 0; i < n; i++) {
            long long a;
            cin >> a;
            sum += a;
        }
        cout << sum * 1000;
    }
    else {
        long long x, sum=0;
        for (long long i = 0; i < n; i++) {
            long long a;
            cin >> a;
            x = a / 1000;
            sum += a % 1000;
        }
        cout << x + sum;
    }
}
```

Task C ()

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

using namespace std;

int f(int a) {
    int q = a;
    int ans = 0;
    for (int i = 0; i < 15; i++) {
        if (q % 2 == 1) ans++;
        q /= 2;
    }
    return ans;
}

vector<vector<int>>> mp;
map<vector<int>, int> mp1;
vector<int> z;

void q(int a) {
    if (a < 10) {
        int k = z.size();
        for (int i = 0; i <= (10-k); i++) {
            q(a + 1);
            z.push_back(a);
        }
        while (z.size() > 0 && z[z.size() - 1] == a) z.pop_back();
    }
    else {
        if (z.size() == 10) {
            mp.push_back(z);
            mp1[z] = mp.size();
        }
    }
}

signed main()
{
    vector<pair<int, int>>> z(3);
    for (int i = 0; i < 3; i++) {
        int a, b;
        cin >> a >> b;
        if (a > b) swap(a, b);
        z[i] = { a, b };
    }
    sort(z.begin(), z.end(), [&](pair<int, int> a, pair<int, int> b) {
        return a.second * a.first > b.second * b.first; });
    map<pair<int, int>, int> m;
    m[z[0]]++;
    m[z[1]]++;
    m[z[2]]++;
    if (z[0] != z[1] && z[0].second == z[1].second) {
        pair<int, int> q = { z[0].first - z[1].first, z[1].second };
        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }
    if (z[0] != z[1] && z[0].first == z[1].second) {
        pair<int, int> q = { z[1].second, z[0].second - z[1].first };
        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }
    if (z[1] != z[2] && z[1].second == z[2].second) {
        pair<int, int> q = { z[1].first - z[2].first, z[2].second };
        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }
    if (z[1] != z[2] && z[1].first == z[2].second) {
        pair<int, int> q = { z[2].second, z[1].second - z[2].first };
    }
}
```

```

        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }

    if (z[0] != z[2] && z[0].second == z[2].second) {
        pair<int, int> q = { z[0].first - z[2].first, z[2].second };
        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }
    if (z[0] != z[2] && z[0].first == z[2].second) {
        pair<int, int> q = { z[2].second, z[0].second - z[2].first };
        if (q.second < q.first) swap(q.second, q.first);
        m[q]++;
    }
}

for (int i = 0; i < 8; i++) {
    int a1, a2, b1, b2;
    if (i == 0) a1 = z[1].first, b1 = z[1].second, a2 = z[2].first, b2 = z[2].second;
    if (i == 1) a1 = z[1].first, b1 = z[1].second, a2 = z[2].second, b2 = z[2].first;
    if (i == 2) a1 = z[1].second, b1 = z[1].first, a2 = z[2].first, b2 = z[2].second;
    if (i == 3) a1 = z[1].second, b1 = z[1].first, a2 = z[2].second, b2 = z[2].first;
    if (i == 4) a1 = z[2].first, b1 = z[2].second, a2 = z[1].first, b2 = z[1].second;
    if (i == 5) a1 = z[2].first, b1 = z[2].second, a2 = z[1].second, b2 = z[1].first;
    if (i == 6) a1 = z[2].second, b1 = z[2].first, a2 = z[1].first, b2 = z[1].second;
    if (i == 7) a1 = z[2].second, b1 = z[2].first, a2 = z[1].second, b2 = z[1].first;

    if (z[0].first == a1 + a2 && b1 + b2 >= z[0].second && b1 <= z[0].second && b2 <= z[0].second)
    {
        pair<int, int> q = { a2, z[0].second - b2 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (z[0].first >= a1 && z[0].first >= a2 && z[0].first < a1 + a2 && b1 + b2 <= z[0].second)
    {
        pair<int, int> q = { b1, z[0].first - a1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
        if (a2 == z[0].first) {
            pair<int, int> q = { z[0].second - b1 - b2, z[0].first };
            if (q.first > q.second) swap(q.first, q.second);
            m[q]++;
        }
    }
    if (b1 == b2 && z[0].first == a1 + a2) {
        pair<int, int> q = { z[0].first, z[0].second - b1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (b1 == b2 && b1 == z[0].second) {
        pair<int, int> q = { z[0].second, z[0].first - b1 - b2 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (a1 == a2 && z[0].second == b1 + b2) {
        pair<int, int> q = { z[0].second, z[0].first - a1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
}

swap(z[0].first, z[0].second);
for (int i = 0; i < 8; i++) {
    int a1, a2, b1, b2;
    if (i == 0) a1 = z[1].first, b1 = z[1].second, a2 = z[2].first, b2 = z[2].second;
    if (i == 1) a1 = z[1].first, b1 = z[1].second, a2 = z[2].second, b2 = z[2].first;
    if (i == 2) a1 = z[1].second, b1 = z[1].first, a2 = z[2].first, b2 = z[2].second;
    if (i == 3) a1 = z[1].second, b1 = z[1].first, a2 = z[2].second, b2 = z[2].first;
    if (i == 4) a1 = z[2].first, b1 = z[2].second, a2 = z[1].first, b2 = z[1].second;
    if (i == 5) a1 = z[2].first, b1 = z[2].second, a2 = z[1].second, b2 = z[1].first;
    if (i == 6) a1 = z[2].second, b1 = z[2].first, a2 = z[1].first, b2 = z[1].second;
    if (i == 7) a1 = z[2].second, b1 = z[2].first, a2 = z[1].second, b2 = z[1].first;

```

```

    if (z[0].first == a1 + a2 && b1 + b2 >= z[0].second && b1 <= z[0].second && b2 <= z[0].
        second) {
        pair<int, int> q = { a2, z[0].second - b2 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (z[0].first >= a1 && z[0].first >= a2 && z[0].first < a1 + a2 && b1 + b2 <= z[0].second
        ) {
        pair<int, int> q = { b1, z[0].first - a1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
        if (a2 == z[0].first) {
            pair<int, int> q = { z[0].second - b1 - b2, z[0].first };
            if (q.first > q.second) swap(q.first, q.second);
            m[q]++;
        }
    }
    if (b1 == b2 && z[0].first == a1 + a2) {
        pair<int, int> q = { z[0].first, z[0].second - b1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (b1 == b2 && b1 == z[0].second) {
        pair<int, int> q = { z[0].second, z[0].first - b1 - b2 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
    if (a1 == a2 && z[0].second == b1 + b2) {
        pair<int, int> q = { z[0].second, z[0].first - a1 };
        if (q.first > q.second) swap(q.first, q.second);
        m[q]++;
    }
}

for (auto &i : m) {
    if (i.first.first > 0) cout << i.first.first << ' ' << i.first.second << '\n';
}
}

```

Task D ()

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

using namespace std;

pair<int, int> print(int a, int b) {
    cout << a << " " << b << endl;
    if (a == -1 && b == -1) exit;
    int q1, q2;
    cin >> q1 >> q2;
    if (q1 == -1 && q2 == -1) exit;
    return { q1, q2 };
}

signed main() {
    int n;
    cin >> n;
    if (n == 1) {
        int a;
        cin >> a;
        print(1, a);
        print(1, a);
    }
    if (n == 3) {
        int a, b, c;
        cin >> a >> b >> c;
        print(-1, -1);
        if (a < b) {
            auto k = print(2, b - a);
            while (true) {
                k = print((k.first) % 2 + 1, k.second);
            }
        }
        else {
            auto k = print(1, a - b);
            while (true) {
                k = print((k.first) % 2 + 1, k.second);
            }
        }
    }
}
```

Task E ()

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

using namespace std;

int f(int a) {
    int q = a;
    int ans = 0;
    for (int i = 0; i < 15; i++) {
        if (q % 2 == 1) ans++;
        q /= 2;
    }
    return ans;
}

vector<vector<int>> mp;
map<vector<int>, int> mp1;
vector<int> z;

void q(int a) {
    if (a <= 10) {
        int k = z.size();
        for (int i = 0; i <= (10 - k); i++) {
            q(a + 1);
            z.push_back(a);
        }
        while (z.size() > 0 && z[z.size() - 1] == a) z.pop_back();
    }
    else {
        if (z.size() == 10) {
            mp.push_back(z);
            mp1[z] = mp.size();
        }
    }
}

signed main() {
    q(0);
    int t;
    cin >> t;
    string s;
    cin >> s;
    if (s[0] == 't') {
        while (t--) {
            int n;
            cin >> n;
            auto z = mp[n - 1];
            for (int i = 0; i < 10; i++) {
                for (int j = 0; j < z[i]; j++) {
                    cout << 1;
                }
                for (int j = z[i]; j < 10; j++) {
                    cout << 0;
                }
                cout << '\n';
            }
        }
    }
    else {
        while (t--) {
            vector<int> z(10);
            for (int i = 0; i < 10; i++) {
                int k = 0;
                for (int j = 0; j < 10; j++) {
                    char a;
                    cin >> a;
                    if (a == '1') z[i]++;
                }
            }
        }
    }
}
```

```
        sort(z.begin(), z.end());  
        cout << mp1[z] << '\n';  
    }  
}
```


Task F ()

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

using namespace std;

int f(int a) {
    int q = a;
    int ans = 0;
    for (int i = 0; i < 15; i++) {
        if (q % 2 == 1) ans++;
        q /= 2;
    }
    return ans;
}

vector<vector<int>> mp;
map<vector<int>, int> mp1;
vector<int> z;

void q(int a) {
    if (a < 10) {
        int k = z.size();
        for (int i = 0; i <= (10-k); i++) {
            q(a + 1);
            z.push_back(a);
        }
        while (z.size() > 0 && z[z.size() - 1] == a) z.pop_back();
    }
    else {
        if (z.size() == 10) {
            mp.push_back(z);
            mp1[z] = mp.size();
        }
    }
}

signed main()
{
    string a, b;
    cin >> a >> b;
    reverse(a.begin(), a.end());
    reverse(b.begin(), b.end());
    while (a.size() != b.size()) {
        if (a.size() < b.size()) a += '0';
        else b += '0';
    }
    string c;
    int d = 0;
    for (int i = 0; i < a.size(); i++) {
        int k = a[i] + b[i] - '0' - '0';
        int r = k + d;
        c += (r%10 + '0');
        if (r < 10) d = 0;
        else d = 1;
    }
    if (d == 1) c += '1';
    reverse(c.begin(), c.end());
    cout << c;
}
```