

Олимпиада СПбГУ по информатике 2022/23 учебного года

A	B	C	D	E	F	Sum
100	100	100	60	52	10	422

Task A ()

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

using namespace std;

signed main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(NULL), cout.tie(NULL);
    vector<int> nums(6);
    for (int i = 0; i < 6; i++) {
        cin >> nums[i];
    }
    vector<int> res(6, 1);
    for (int i = 0; i < 6; i++) {
        int last = 0;
        for (int y = 0; y < 6; y++) {
            if (nums[y] == 1) {
                last = y;
            }
        }
        res[last] = i + 1;
        nums[last] = 100000;
        for (int y = last + 1; y < 6; y++) {
            nums[y]--;
        }
    }
    for (int elem : res) {
        cout << elem << " ";
    }
}
```

Task B ()

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

#define int long long

using namespace std;

const int N = 2e5 * 2 + 1, inf = 2e9 + 11;

signed main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(NULL), cout.tie(NULL);
    string s;
    cin >> s;
    int n, sum = 0;
    cin >> n;
    vector<int> a(n);
    for (int i = 0; i < n; i++) {
        cin >> a[i];
        sum += a[i];
    }
    if (s == "first") {
        string x = to_string(sum / n) + "000";
        int ost = sum % n;
        while (x.size() < 7) {
            x = "0" + x;
        }
        x = "1" + x;
        string res = to_string(ost) + x;
        cout << stoll(res) << "\n";
    }
    else {
        sum = 0;
        for (int i = 0; i < n; i++) {
            sum += a[i] % 1000;
        }
        int x = ((a[0] - (a[0] % 1000)) % 100000000) / 1000;
        sum += x * n;
        int ost = a[0] / 100000000;
        sum += ost;
        cout << sum << "\n";
    }
}
```

Task C ()

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

#define int long long

using namespace std;

const int N = 2e5 * 2 + 1, inf = 2e9 + 11;

bool cmp(const pair<int, int>& a, const pair<int, int>& b) {
    return (a.first * a.second) < (b.first * b.second);
}

signed main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(NULL), cout.tie(NULL);
    vector<pair<int, int>> have(3);
    vector<pair<int, int>> res(3);
    for (int i = 0; i < 3; i++) {
        cin >> have[i].first >> have[i].second;
        res[i] = have[i];
    }
    sort(have.begin(), have.end(), cmp);
    for (int i = 0; i < 2; i++) {
        for (int y = i + 1; y < 3; y++) {
            if (have[i].first * have[i].second == have[y].first * have[y].second) {
                continue;
            }
            if (have[i].first == have[y].first) {
                res.push_back({ have[y].first, have[y].second - have[i].second });
            }
            if (have[i].first == have[y].second) {
                res.push_back({ have[y].second, have[y].first - have[i].second });
            }
            if (have[i].second == have[y].first) {
                res.push_back({ have[y].first, have[y].second - have[i].first });
            }
            if (have[i].second == have[y].second) {
                res.push_back({ have[y].second, have[y].first - have[i].first });
            }
        }
    }
    if (have[0].first * have[0].second + have[1].first * have[1].second < have[2].first * have[2].second) {
        pair<int, int> main = have[2];
        pair<int, int> one = have[0];
        pair<int, int> two = have[1];
        for (int q1 = 0; q1 < 2; q1++) {
            swap(one, two);
            for (int t1 = 0; t1 < 2; t1++) {
                swap(main.first, main.second);
                for (int i1 = 0; i1 < 2; i1++) {
                    swap(one.first, one.second);
                    for (int y1 = 0; y1 < 2; y1++) {
                        swap(two.first, two.second);
                        if (max(one.first, two.first) > main.first) continue;
                        if (max(one.second, two.second) > main.second) continue;
                        if (two.second == main.second - one.second) {
                            if (two.first == one.first) {
                                res.push_back({ main.second, main.first - one.first });
                            }
                            if (two.first == main.first) {
                                res.push_back({ main.first - one.first, main.second - two.second });
                            }
                        }
                        if (two.first + one.first >= main.first) {
                            res.push_back({ main.first - two.first, main.second - one.second });
                            res.push_back({ main.first - one.first, main.second - two.second });
                        }
                    }
                }
            }
        }
    }
}
```


Task D ()

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

using namespace std;

const int N = 51 * 2 + 11, inf = 2e9 + 11;

int dp[N][N][N];

signed main()
{
    int n;
    cin >> n;
    vector<int> have(n);
    for (int i = 0; i < n; i++) {
        cin >> have[i];
    }
    if (n == 1) {
        cout << 1 << " " << have[0] << "\n";
        int t, q;
        cin >> t >> q;
        if (t == -1) {
            return 0;
        }
        cout << 1 << " " << have[0] << "\n";
        cin >> t >> q;
        if (t == -1) {
            return 0;
        }
        return 0;
    }
    for (int a = 1; a <= have[0]; a++) dp[a][0][0] = 1;
    for (int a = 1; a <= have[1]; a++) dp[0][a][0] = 1;
    for (int a = 1; a <= have[2]; a++) dp[0][0][a] = 1;

    for (int a = 0; a <= have[0] * 2 + 2; a++) {
        for (int b = 0; b <= have[1] * 2 + 2; b++) {
            for (int c = 0; c <= have[2] * 2 + 2; c++) {
                if (dp[a][b][c] == 1) continue;
                if (a <= have[0]) {
                    for (int i = 0; i < a; i++) {
                        dp[a][b][c] |= !dp[i][b][c];
                    }
                }
                if (b <= have[1]) {
                    for (int i = 0; i < b; i++) {
                        dp[a][b][c] |= !dp[a][i][c];
                    }
                }
                if (c <= have[2]) {
                    for (int i = 0; i < c; i++) {
                        dp[a][b][c] |= !dp[a][b][i];
                    }
                }
                if (a >= have[0] + 1) {
                    dp[a][b][c] |= !dp[have[0]][b][c];
                }
                if (b >= have[1] + 1) {
                    dp[a][b][c] |= !dp[a][have[1]][c];
                }
                if (c >= have[2] + 1) {
                    dp[a][b][c] |= !dp[a][b][have[2]];
                }

                for (int i = have[0] + 1; i < a; i++) {
                    dp[a][b][c] |= !dp[i][b][c];
                }
                for (int i = have[1] + 1; i < b; i++) {
                    dp[a][b][c] |= !dp[a][i][c];
                }
            }
        }
    }
}
```

```

        }
        for (int i = have[2] + 1; i < c; i++) {
            dp[a][b][c] |= !dp[a][b][i];
        }
    }
}
int a = have[0], b = have[1], c = have[2];
while (1) {
    if (!dp[a][b][c]) {
        cout << "-1_-1\n";
        return 0;
    }
    int flag = 0;
    pair<int, int> ans;
    if (a <= have[0]) {
        for (int i = 0; i < a; i++) {
            if (!dp[i][b][c]) {
                ans = { 1, a - i };
                flag = 1;
                break;
            }
        }
    }
    if (!flag && b <= have[1]) {
        for (int i = 0; i < b; i++) {
            if (!dp[a][i][c]) {
                ans = { 2, b - i };
                flag = 1;
                break;
            }
        }
    }
    if (!flag && c <= have[2]) {
        for (int i = 0; i < c; i++) {
            if (!dp[a][b][i]) {
                ans = { 3, c - i };
                flag = 1;
                break;
            }
        }
    }
    if (!flag && a >= have[0] + 1) {
        if (!dp[have[0]][b][c]) {
            flag = 1;
            ans = { 1, 0 };
        }
    }
    if (!flag && b >= have[1] + 1) {
        if (!dp[a][have[1]][c]) {
            flag = 1;
            ans = { 2, 0 };
        }
    }
    if (!flag && c >= have[2] + 1) {
        if (!dp[a][b][have[2]]) {
            flag = 1;
            ans = { 3, 0 };
        }
    }
    if (!flag && a > have[0] + 1) {
        for (int i = have[0] + 1; i < a; i++) {
            if (!dp[i][b][c]) {
                ans = { 1, a - i };
                flag = 1;
                break;
            }
        }
    }
    if (!flag && b > have[1] + 1) {
        for (int i = have[1] + 1; i < b; i++) {
            if (!dp[a][i][c]) {
                ans = { 2, b - i };
            }
        }
    }
}

```

```

        flag = 1;
        break;
    }
}
}
if (!flag && c > have[2] + 1) {
    for (int i = have[2] + 1; i < c; i++) {
        if (!dp[a][b][i]) {
            ans = { 3, c - i };
            flag = 1;
            break;
        }
    }
}
cout << ans.first << " " << ans.second << "\n";
int q, t;
cin >> q >> t;
if (q == -1) return 0;
if (ans.first == 1) {
    if (ans.second == 0) a = have[0];
    else a -= ans.second;
}
if (ans.first == 2) {
    if (ans.second == 0) b = have[1];
    else b -= ans.second;
}
if (ans.first == 3) {
    if (ans.second == 0) c = have[2];
    else c -= ans.second;
}
if (q == 1) {
    if (t == 0) a = have[0];
    else a -= t;
}
if (q == 2) {
    if (t == 0) b = have[1];
    else b -= t;
}
if (q == 3) {
    if (t == 0) c = have[2];
    else c -= t;
}
if (a + b + c == 0) return 0;
}
}

```

Task E ()

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

#define int long long

using namespace std;

const int N = 2e5 * 2 + 1, inf = 2e9 + 11;

int res = 0;

vector<int> all[N];

void did(int have, int now, vector<int> nums) {
    if (have == 0) {
        all[res] = nums;
        res++;
        // for (int elem : nums) {
        //     cout << elem << " ";
        // }
        // cout << "\n";
        return;
    }
    for (int i = now; i <= 10; i++) {
        nums.push_back(i);
        did(have - 1, i, nums);
        nums.pop_back();
    }
}

signed main()
{
    ios_base::sync_with_stdio(0);
    cin.tie(NULL), cout.tie(NULL);
    did(10, 0, {});
    int t;
    cin >> t;
    string s;
    cin >> s;
    while (t--) {
        if (s == "transmit") {
            int num;
            cin >> num;
            for (int elem : all[num - 1]) {
                for (int i = 0; i < 10; i++) {
                    if (i < elem) {
                        cout << "1";
                    } else {
                        cout << "0";
                    }
                }
                cout << "\n";
            }
        } else {
            vector<int> nums(10, 0);
            for (int i = 0; i < 10; i++) {
                string have;
                cin >> have;
                for (int y = 0; y < 10; y++) {
                    nums[i] += have[y] - '0';
                }
            }
            sort(nums.begin(), nums.end());
            for (int i = 0; i < res; i++) {
                if (all[i] == nums) {
                    cout << i + 1 << "\n";
                    break;
                }
            }
        }
    }
}
```


}
}
}
}

Task F ()