

Олимпиада СПбГУ по информатике 2020/21 учебного года

A	B	C	D	E	F	Sum
100	100	100	100	55	15	470

Task A ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

void solve() {
    int n;
    cin >> n;
    if (n < 10) {
        cout << n;
        return;
    }
    n -= 10;
    if (n % 9 == 0) {
        cout << 0;
        return;
    }
    else {
        cout << n % 9 + 1;
    }
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}
```

Task B ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

void solve() {
    int n, k;
    cin >> n >> k;
    vector<char> cnts(26, false);
    string s;
    cin >> s;
    int cur_cnt = 0;
    int cur_size = 0;
    int ans = 0;
    for (char i: s) {
        if (!cnts[i - 'a']) {
            cur_cnt++;
        }
        cur_size++;
        if (cur_cnt == 4 || cur_size > k) {
            ans++;
            cur_size = 1;
            cur_cnt = 1;
            cnts.assign(26, false);
        }
        cnts[i - 'a'] = true;
    }
    ans++;
    cout << ans;
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}
```

Task C ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

void solve() {
    int n, x, y;
    cin >> n >> x >> y;
    vector<int> weights(n);
    vector<int> vols(n);
    ll sum = 0;
    for (int &i: vols) {
        cin >> i;
    }
    for (int &i: weights) {
        cin >> i;
        sum += i;
    }
    vector<vector<int>> backup(n, vector<int>(x + 1, -1));
    vector<int> last(x + 1, -1);
    last[0] = 0;
    vector<int> cur(x + 1, -1);
    for (int i = 0; i < n; i++) {
        for (int j = 0; j <= x; j++) {
            if (last[j] != -1) {
                cur[j] = last[j];
                backup[i][j] = 0;
            }
            if (j >= vols[i] && last[j - vols[i]] != -1 && cur[j] < last[j - vols[i]] + weights[i]) {
                cur[j] = last[j - vols[i]] + weights[i];
                backup[i][j] = 1;
            }
        }
        for (int j = 0; j <= x; j++) {
            last[j] = cur[j];
        }
    }
    int vol = 0, weight = 0;
    for (int i = 0; i <= x; i++) {
        if (cur[i] > weight) {
            weight = cur[i];
            vol = i;
        }
    }
    if (sum - weight > y) {
        cout << -1;
        return;
    }
    vector<char> taken;
    for (int i = n - 1; i >= 0; i--) {
        if (backup[i][vol]) {
            vol -= vols[i];
            taken.push_back('x');
        }
        else {
            taken.push_back('y');
        }
    }
    for (int i = n - 1; i >= 0; i--) {
        cout << taken[i];
    }
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}
```

Task D ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

void solve() {
    int n;
    cin >> n;
    string s;
    cin >> s;
    vector<int> a;
    int cur_num;
    for (char i: s) {
        if (i == ')' || i == '(') {
            cur_num = 0;
        }
        else {
            cur_num = 1;
        }
        if (a.size() && a.back() == cur_num) {
            a.pop_back();
        }
        else {
            a.push_back(cur_num);
        }
    }
    cout << a.size() / 2;
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}
```

Task E ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

vector <vector <vector <int>>> ans(11, vector <vector <int>> (11, vector <int> (11, -1)));

void place(int i, int j, int k) {
    if (ans[i][j][k] != -1) {
        return;
    }
    for (int cur = 1; cur <= 10; cur++) {
        if (cur == i || cur == j || cur == k) {
            continue;
        }
        if (i != ans[cur][j][k] && j != ans[i][cur][k] && k != ans[i][j][cur]) {
            ans[i][j][k] = cur;
            ans[i][k][j] = cur;
            ans[k][i][j] = cur;
            ans[k][j][i] = cur;
            ans[j][k][i] = cur;
            ans[j][i][k] = cur;
            break;
        }
    }
}

void solve() {
    for (int i = 1; i <= 10; i++) {
        for (int j = 1; j <= 10; j++) {
            if (j == i) {
                continue;
            }
            for (int k = 1; k <= 10; k++) {
                if (k == j || k == i) {
                    continue;
                }
                place(i, j, k);
            }
        }
    }
    string com;
    cin >> com;
    int t;
    cin >> t;
    int n;
    int k;
    vector <char> a;
    vector <int> inf;
    ll sum = 0;
    int cur;
    ll cur_sum;
    int fir, sec, thir, four;
    if (com == "add") {
        for (int smth = 0; smth < t; smth++) {
            cin >> n >> k;
            if (n == 10 && k == 3) {
                cin >> fir >> sec >> thir;
                cout << ans[fir][sec][thir] << '\n';
                continue;
            }
            a.resize(n + 1);
            a.assign(n + 1, false);
            sum = 0;
            for (int i = 0; i < k; i++) {
                cin >> cur;
                sum += cur;
                a[cur] = true;
            }
            sum %= rand();
            sum %= n;
        }
    }
}
```

```

        sum++;
        while (a[sum]) {
            sum = sum % n + 1;
        }
        cout << sum << '\n';
    }
}
else {
    for (int smth = 0; smth < t; smth++) {
        cin >> n >> k;
        if (n == 10 && k == 3) {
            cin >> fir >> sec >> thir >> four;
            if (ans[fir][sec][thir] == four) {
                cout << fir << '␣' << sec << '␣' << thir << '\n';
            }
            else if (ans[sec][thir][four] == fir) {
                cout << sec << '␣' << thir << '␣' << four << '\n';
            }
            else if (ans[thir][four][fir] == sec) {
                cout << thir << '␣' << four << '␣' << fir << '\n';
            }
            else if (ans[four][fir][sec] == thir) {
                cout << four << '␣' << fir << '␣' << sec << '\n';
            }
            continue;
        }
        a.resize(n + 1);
        a.assign(n + 1, false);
        inf.resize(k + 1);
        sum = 0;
        for (int &i: inf) {
            cin >> i;
            a[i] = true;
            sum += i;
        }
        cur = rand();
        for (int i = 0; i < inf.size(); i++) {
            a[inf[i]] = false;
            cur_sum = sum - inf[i];
            cur_sum %= cur;
            cur_sum %= n;
            cur_sum++;
            while (a[cur_sum]) {
                cur_sum = cur_sum % n + 1;
            }
            if (cur_sum == inf[i]) {
                cur = inf[i];
                break;
            }
            a[inf[i]] = true;
        }
        for (int i = 0; i < inf.size(); i++) {
            if (inf[i] != cur) {
                cout << inf[i] << '␣';
            }
        }
        cout << '\n';
    }
}
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}

```

Task F ()

```
#include <bits/stdc++.h>

using namespace std;
typedef long long ll;
typedef long double ld;

void solve() {
    cout << 4 << '\n';
    cout << 0 << '\n' << 0 << '\n';
    cout << 0 << '\n' << 2 << '\n';
    cout << 2 << '\n' << 2 << '\n';
    cout << 2 << '\n' << 0 << '\n';
    vector <pair <int, int> > a;
    a.push_back({1, 2});
    a.push_back({-1, 2});
    a.push_back({1, -2});
    a.push_back({-1, -2});
    a.push_back({2, 0});
    a.push_back({-2, 0});
    int n;
    cin >> n;
    for (int i = 0; i < n; i++) {
        cout << a[i].first << '\n' << a[i].second << '\n';
        if (i == 5) {
            return;
        }
    }
}

int main() {
    cin.tie(0);
    cout.tie(0);
    ios_base::sync_with_stdio(0);
    solve();
}
```