

Рјешења задатака  
Регионално Такмичење из Информатике  
21.03.2026

## 1 Бродови

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

using namespace std;

int main() {
    int t;
    scanf("%d", &t);
    while (t--) {
        int n;
        long long k;
        scanf("%d %lld", &n, &k);
        vector<long long> a(n);
        for (int i = 0; i < n; i++) scanf("%lld", &a[i]);

        deque<long long> dq(a.begin(), a.end());
        bool front = true;
        for (long long i = 0; i < k && !dq.empty(); i++) {
            if (front) {
                dq.front()--;
                if (dq.front() == 0) dq.pop_front();
            } else {
                dq.back()--;
                if (dq.back() == 0) dq.pop_back();
            }
            front = !front;
        }

        printf("%d\n", n - (int)dq.size());
    }
    return 0;
}
```

## 2 ИНСТИТУТ

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

using namespace std;

int longestConsecutive(vector<int>& a) {
    if (a.empty()) return 0;

    sort(a.begin(), a.end());

    int best = 1;
    int curr = 1;

    for (int i = 1; i < a.size(); i++) {
        if (a[i] == a[i-1] + 1) {
            curr++;
            best = max(best, curr);
        } else if (a[i] != a[i-1]) {
            curr = 1;
        }
    }

    return best;
}

int main() {
    long long n, m;
    cin >> n >> m;

    int k;
    cin >> k;

    vector<int> a(k);
    for (int i = 0; i < k; i++)
        cin >> a[i];

    int l;
    cin >> l;

    vector<int> b(l);
    for (int i = 0; i < l; i++)
        cin >> b[i];

    long long height = longestConsecutive(a) + 1;
    long long width = longestConsecutive(b) + 1;

    cout << height * width << endl;

    return 0;
}
```

### 3 Пирамида

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;

int max_gold(int n, const vector<vector<int>>& tr) {
    vector<vector<int>> dp(n, vector<int>(n));
    dp[0][0] = tr[0][0];
    for(int i=1; i<n; i++) {
        for(int j=0; j<=i; j++) {
            if (j > 0) dp[i][j] = dp[i-1][j-1] + tr[i][j];
            if (j < i) dp[i][j] = max(dp[i][j], dp[i-1][j] + tr[i][j]);
        }
    }
    int res = 0;
    for(int i=0; i<n; i++) {
        res = max(res, dp[n-1][i]);
    }
    return res;
}

int main()
{
    int n; cin >> n;
    vector<vector<int>> tr(n, vector<int>(n));
    for(int i=0; i<n; i++) {
        for(int j=0; j<=i; j++) {
            cin >> tr[i][j];
        }
    }

    cout << max_gold(n, tr) << '\n';

    return 0;
}
```

## 4 Базен

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

using namespace std;

using ll = long long;

int main() {
    ll h, c, t;
    cin >> h >> c >> t;

    if (h + c - 2 * t >= 0)
        cout << 2 << endl;
    else {
        ll a = h - t;
        ll b = 2 * t - c - h;
        ll k = a / b;

        ll val1 = abs( (k*(h + c) + h) - t * (2 * k + 1) );
        ll val2 = abs( ((k + 1) * (h + c) + h) - t * (2 * k + 3) );
        printf("%lld\n", val1 * (2 * k + 3) <= val2 * (2 * k + 1) ? 2 * k + 1 : 2 * k + 3);
    }

    return 0;
}
```

## 5 Неко је побјегао...

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

using namespace std;

vector<vector<int>> graph;
vector<int> cc;

void buildCC(int startNode, int ccIdx) {
    vector<int> stack = {startNode};
    cc[startNode] = ccIdx;
    while (!stack.empty()) {
        int currentNode = stack.back();
        stack.pop_back();
        for (int nextNode : graph[currentNode]) {
            if (cc[nextNode] == -1) {
                cc[nextNode] = ccIdx;
                stack.push_back(nextNode);
            }
        }
    }
}

int main() {
    int N, M;
    cin >> N >> M;

    graph.resize(N + 1, vector<int>());
    cc.resize(N + 1, -1);

    for (int i = 0; i < M; i++) {
        int u, v;
        cin >> u >> v;
        graph[u].push_back(v);
        graph[v].push_back(u);
    }

    int ccIdx = 1;
    for (int node = 1; node <= N; node++) {
        if (cc[node] == -1) {
            buildCC(node, ccIdx);
            ccIdx++;
        }
    }

    set<array<int, 2>> pairsCC;

    int K;
    cin >> K;
    vector<array<int, 2>> pairs(K, array<int, 2>());
    for (int i = 0; i < K; i++) {
        cin >> pairs[i][0] >> pairs[i][1];

        int pCC0 = cc[pairs[i][0]];
        int pCC1 = cc[pairs[i][1]];

        pairsCC.insert({pCC0, pCC1});
    }
}
```

```

        pairsCC.insert({pCC1, pCC0});
    }

    int Q;
    cin >> Q;
    while (Q--) {
        int p, q;
        cin >> p >> q;

        int pCC = cc[p];
        int qCC = cc[q];
        if (pairsCC.count({pCC, qCC}))
            cout << "Ne" << endl;
        else
            cout << "Da" << endl;
    }

    return 0;
}

```