Олимпиада СПбГУ
по информатике
2023/24 учебного года

24fin0077 – Ваньжа Кирилл Евгеньевич

A	B	C	D	E	F	sum
100	100	40	100	40	91	471

Task A

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

int64_t n, a, b;

struct entry {
    int64_t val = -1;
};

unordered_map<int64_t, entry> mp;

int64_t f(int64_t n, int64_t need) {
    if (~mp[n - need].val)
        return mp[n - need].val;

    if (n < need)
        return mp[n - need].val = 0;
    if (n == need)
        return mp[n - need].val = 1;
    return mp[n - need].val = f(n - a, need) + f(n - b, need);// - f(n -
    lcm(a, b), need);
}

int main() {
```

```
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    cin >> a >> b >> n;

    int64_t res = 0;
    for (int64_t i = 0; i < a; i++) {
        res += f(n - i, a - i);
    }

    for (int64_t i = 0; i < b; i++) {
        res += f(n - i, b - i);
    }

    cout << res << '\n';

    return 0;
}
```

Task B

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

int dist(int x1, int y1, int z1, int x2, int y2, int z2) {
    return (abs(x1 - x2) + abs(y1 - y2) + abs(z1 - z2)) / 2;
}

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int n;
    cin >> n;
    int x1, y1, z1;
    cin >> x1 >> y1 >> z1;
    int x2, y2, z2;
    cin >> x2 >> y2 >> z2;

    int res = dist(x1, y1, z1, x2, y2, z2);

    int x3, y3, z3;

    x3 = (x2 > 0 ? -1 : 1) * 2 * n + x2;
    y3 = (x2 > 0 ? 1 : -1) * n + y2;
    z3 = (x2 > 0 ? 1 : -1) * n + z2;
    // swap(y3, z3);
    res = min(res, dist(x1, y1, z1, x3, y3, z3));
```

```
    x3 = (y2 > 0 ? 1 : -1) * n + x2;
    y3 = (y2 > 0 ? -1 : 1) * 2 * n + y2;
    z3 = (y2 > 0 ? 1 : -1) * n + z2;
    // swap(x3, z3);

    res = min(res, dist(x1, y1, z1, x3, y3, z3));

    x3 = (z2 > 0 ? 1 : -1) * n + x2;
    y3 = (z2 > 0 ? 1 : -1) * n + y2;
    z3 = (z2 > 0 ? -1 : 1) * 2 * n + z2;
    // swap(x3, y3);

    res = min(res, dist(x1, y1, z1, x3, y3, z3));

    cout << res << '\n';

    return 0;
}
```

Task C

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    string cmd;
    cin >> cmd;
    if (cmd[0] == 'f') {
        int n;
        cin >> n;
        string str;
        cin >> str;
        cout << str << '\n';
    } else {
        int n;
        cin >> n;
        string str, h;
        cin >> str >> h;

        for (int s = 0; s < n; s++) {
            bool res = true;
            for (int i = 0; i < n && res; i++) {
                res &= (str[(i + s) % n] == h[i]);
            }
```

```
            if (res) {
                cout << "YES\n";
                return 0;
            }
        }

        cout << "NO\n";
    }

    return 0;
}
```

Task D

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

int curr = 0;

char move(int a, int b) {
    cout << a << ' ' << b << '\n';
    cout.flush();
    char c;
    cin >> c;
    curr++;
    if (c == '!')
        exit(0);
    return c;
}

int main() {
    ios_base::sync_with_stdio(false);
    //cin.tie(nullptr);

    int n;
    cin >> n;

    while (true) {
        char c = move(1, 1);
        if (curr % n != 0 && c == '>') {
            move(1, n);
        }
    }

    return 0;
}
```

Task E

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

vector<vector<int>> g;


int main() {
    ios_base::sync_with_stdio(false);
    //cin.tie(nullptr);

    int n, q;
    cin >> n >> q;

    g.resize(n);
    int sum = 0;
    for (int i = 0; i < n; i++) {
        int sz;
        cin >> sz;
        sum += sz;
        g[i].resize(sz);
        for (auto &e : g[i]) {
            cin >> e;
            e--;
        }
    }

    vector<int> pos(n, 0);
    int v = 0;
    vector<int> path;
    for (int i = 0; i < 5e7; i++) {
        path.push_back(v);
        v = g[v][(pos[v]++) % g[v].size()];
    }

    vector<int> path2;
    for (int i = 0; i < sum; i++) {
        path2.push_back(v);
        v = g[v][(pos[v]++) % g[v].size()];
    }

    while (q--) {
        int64_t x;
        cin >> x;
        if (x < path.size()) {
            cout << path[x] + 1 << ' ';
        } else {
```

```
            cout << path2[(x - path.size()) % sum] + 1 << ' ';
        }
    }

    return 0;
}
```

Task F

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

using namespace std;

typedef complex<double> point;
#define x() real()
#define y() imag()

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(nullptr);

    int h, w;
    cin >> h >> w;
    vector<string> arr(h);
    for (auto &e : arr)
        cin >> e;

    vector<tuple<int, int, int, int, int>> v;
    for (int i = 0; i < h; i++) {
        for (int j = 0; j < w; j++) {
            v.push_back({ (arr[i][j] - '0') * (arr[i][(j + 1) % w] - '0')
    , i, j, i, (j + 1) % w });
            v.push_back({ (arr[i][j] - '0') * (arr[(i + 1) % h][j] - '0')
    , i, j, (i + 1) % h, j });
        }
    }

    sort(v.rbegin(), v.rend());

    vector<string> res(h, "");
    for (auto &e : res)
        for (int i = 0; i < w; i++)
            e += '.';

    int sum = 0;

    for (auto e : v) {
        auto [ m, a, b, c, d ] = e;
        if (res[a][b] != '.' || res[c][d] != '.')
            continue;

        sum += m;
        if (a == c) {
```

```
            res[a][b] = 'L';
            res[c][d] = 'R';
        } else {
            res[a][b] = 'T';
            res[c][d] = 'B';
        }
    }

    cout << sum << '\n';
    for (auto &e : res)
        cout << e << '\n';

    return 0;
}
```