# Олимпиада СПбГУ
# по информатике
# 2023/24 учебного года

24fin0128 – Густов Ян Евгеньевич

| A | B | C | D | E | F | sum |
|---|---|---|---|---|---|---|
| 100 | 100 | 100 | 100 | 40 | . | 440 |

## Task A

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

using namespace std;

#define int long long

int C[100][100];

signed main() {
    C[0][0] = 1;
    for (int i = 1; i < 100; ++i) {
        C[i][0] = C[i][i] = 1;
    }
    for (int i = 1; i < 100; ++i) {
        for (int j = 1; j < i; ++j) {
            C[i][j] = C[i - 1][j] + C[i - 1][j - 1];
        }
    }
    int a, b, n;
    cin >> a >> b >> n;
    if (a > b) {
        swap(a, b);
    }
    long long ans = 0;
    for (int x = 0; x <= n; ++x) {
        for (int y = 0; y <= n; ++y) {
            if (a * x + b * y == n) {
```

```
                if (x > 0) {
                    ans += a * C[x + y - 1][x - 1];
                }
                if (y > 0) {
                    ans += b * C[x + y - 1][y - 1];
                }
            }
        }
    }
    cout << ans;
}
```

## Task B

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

using namespace std;

#define int long long

int a[6];

int d() {
    return (abs(a[0] - a[3]) + abs(a[1] - a[4]) + abs(a[2] - a[5])) / 2;
}

signed main() {
    int n;
    cin >> n;
    for (int i = 0; i < 6; ++i) {
        cin >> a[i];
    }
    int b[7][6] = {{0,0,0,0,0,0},
                   {2 * n, -n, -n, 0,0,0},
                   {-n, 2*n, -n, 0,0,0},
                   {-n,-n, 2*n, 0,0,0},
                   {0,0,0,2*n,-n,-n},
                   {0,0,0,-n,2*n,-n},
                   {0,0,0,-n,-n,2*n}};
    int ans = d();
    for (int i = 0; i < 7; ++i) {
        for (int j = 0; j < 6; ++j) {
            a[j] += b[i][j];
        }
        ans = min(ans, d());
        for (int j = 0; j < 6; ++j) {
            a[j] -= b[i][j];
        }
    }
    cout << ans;
}
```

## Task C

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

using namespace std;

#define int long long

const int N = 2e5;
const int p = 31;
const int mod = 1e9 + 9;

int powp[N];

int h[N];

string s;
int n;

void calc_h() {
    h[0] = 0;
    for (int i = 1; i <= n; ++i) {
        h[i] = h[i - 1] * p % mod + (s[i - 1] - '0' + 1);
        h[i] %= mod;
    }
}

int get_h(int shift) {
    return (h[shift] + ((h[n] - h[shift] * powp[n - shift] % mod + mod) %
     mod) * powp[shift] % mod) % mod;
}

signed main() {
    powp[0] = 1;
    for (int i = 1; i < N; ++i) {
        powp[i] = powp[i - 1] * p % mod;
    }
    string type;
    cin >> type;
    cin >> n;
    cin >> s;
    calc_h();
    int now = 0;
    for (int i = 1; i <= n; ++i) {
        now = max(now, get_h(i));
    }
    if (type == "first") {
        cout << now;
    }
    else {
        int ans;
```

```
        cin >> ans;
        cout << (ans == now ? "YES" : "NO");
    }
}
```

## Task D

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

using namespace std;

#define int long long

const int N = 1000;

int pos[N];
int where[N];
int el[N];

int n;

bool que(int i, int j) {
    cout << where[i] + 1 << " " << j << endl;
    swap(pos[where[i]], pos[j - 1]);
    int first = pos[0];
    for (int i = 0; i + 1 < n; ++i) {
        pos[i] = pos[i + 1];
        where[pos[i]] = i;
    }
    pos[n - 1] = first;
    where[pos[n - 1]] = n - 1;
    char symbol;
    cin >> symbol;
    if (symbol == '!') {
        exit(0);
    }
    return (symbol == '<');
}

bool cmp(int i, int j) {
    que(j, 2);
    return que(i, 2);
}

signed main() {
    cin >> n;
    for (int i = 0; i < n; ++i) {
        el[i] = i;
        pos[i] = i;
        where[i] = i;
    }
```

```
    sort(el, el + n, cmp);
    for (int i = n - 1; i >= 0; --i) {
        que(el[i], n);
    }
}
```

## Task E

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

using namespace std;

#define int long long

const int N = 1e7;
const int M = 1e5 + 5;

int s[M];
vector<int> e[M];
int pred[N];
int ans[N];
int par[M];
int sz[M];

bool is_ok(int i) {
    if (i == 0) {
        return true;
    }
    return (e[i][(s[i] - 1 + e[i].size()) % e[i].size()]) == par[i];
}

int dfs(int i, int p = -1) {
    sz[i] = 1;
    for (int j : e[i]) {
        if (j != p) {
            par[j] = i;
            sz[i] += dfs(j, i);
        }
    }
    return sz[i];
}

signed main() {
    int n, q;
    cin >> n >> q;
    for (int i = 0; i < n; ++i) {
        int k;
        cin >> k;
        for (int j = 0; j < k; ++j) {
            int a;
```

```
        cin >> a;
        e[i].push_back(a - 1);
    }
}
if (n <= 1000) {
    dfs(0);
    int ok = 0;
    for (int i = 0; i < n; ++i) {
        ok += is_ok(i);
    }
    int now = 0;
    int t = 0;
    pred[0] = 0;
    while (ok != n || now != 0) {
        int p = s[now];
        ok -= is_ok(now);
        s[now] = (s[now] + 1) % e[now].size();
        ok += is_ok(now);
        now = e[now][p];
        ++t;
        pred[t] = now;
    }
    int predp = t;
    t = 0;
    while (t != 2 * n - 2) {
        int p = s[now];
        ok -= is_ok(now);
        s[now] = (s[now] + 1) % e[now].size();
        ok += is_ok(now);
        now = e[now][p];
        ++t;
        ans[t] = now;
    }
    while (q--) {
        int que;
        cin >> que;
        if (que <= predp) {
            cout << pred[que] + 1 << endl;
        } else {
            cout << ans[(que - predp) % t] + 1 << endl;
        }
    }
}
else if (q <= 10) {
    dfs(0);
    int ok = 0;
    for (int i = 0; i < n; ++i) {
        ok += is_ok(i);
    }
    int t = 0;
    int now = 0;
    while (q--) {
        int que;
```

```
            cin >> que;
            while (t != que) {
                if (ok == n || (now != 0 && is_ok(now) && t + 2 * sz[now]
 - 1 < que)) {
                    t += 2 * sz[now] - 2;
                }
                else {
                    int p = s[now];
                    ok -= is_ok(now);
                    s[now] = (s[now] + 1) % e[now].size();
                    ok += is_ok(now);
                    now = e[now][p];
                    ++t;
                }
            }
        }
    }
    else {
        int now = 0;
        int t = 0;
        while (q--) {
            int que;
            cin >> que;
            while (t != que) {
                int p = s[now];
                s[now] = (s[now] + 1) % e[now].size();
                now = e[now][p];
                ++t;
            }
            cout << now + 1 << endl;
        }
    }
}
```