**Задача A**

#include <iostream>

using namespace std;

int main(){ int a, b, c, d;

cin >> a >> b >> c >> d;

cout << a + b + c - d;

}

**Задача B**

#include <iostream>

using namespace std;

int main() {

int a, c; cin >> a >> c;

cout << c + c - a;

}

**Задача C**

#include <iostream>

using namespace std;

int main() {

int a, b, x, y; cin >> a >> b >> x >> y;

long long total = a + b \* 2, score = (a - x) + (b - y) \* 2;

cout << (score \* 100 >= total \* 51 ? "YES" : "NO");

}

**Задача D**

#include <iostream>

using namespace std;

int main() {

long long db, tb, wb, dm, tm, wm, M, B, dt, tt, T, minVal;

cin >> db >> tb >> wb;B = db + tb + wb;

cin >> dm >> tm >> wm; M = dm + tm + wm;

cin >> dt >> tt;T = dt + tt;

minVal = min({ B, M, T });

if (B == minVal) cout << "Bus";

else if (M == minVal) cout << "Metro";

else cout << "Taxi";

}

**Задача E**

#include <iostream>

using namespace std;

int main() {

int a, b, n, m, k, d = 1, res = 0; cin >> a >> b >> n >> m >> k;

while (d <= k) {

for (int i = 0; i < a && d <= k; ++i) {

if (d % n == 0) res += 0;

else if (d % m == 0) res += 2;

else res += 1; ++d;

}d += b;}

cout << res;

}

**Задача F**

#include <iostream>

#define for for(int i = 0; i < n\_m; i++)

using namespace std;

long long a[(int)200'000], b[(int)200'000];

int main() {

int n, m, n\_m; cin >> n >> m;

n\_m = n; for cin >> a[i]; n\_m = m; for cin >> b[i];

long long e, scoreA = 0, scoreB = 0; cin >> e;

n\_m = n; for if (e \* (e + 1) / 2 >= a[i]) scoreA += a[i];

else { scoreA += e \* (e + 1) / 2; break; }

n\_m = m; for if (e \* (e + 1) / 2 >= b[i]) scoreB += b[i];

else { scoreB += e \* (e + 1) / 2; break; }

if (scoreA == scoreB) cout << "Draw\n" << scoreA << ':' << scoreB;

else if (scoreA > scoreB) cout << "Danya\n" << scoreA << ':' << scoreB;

else cout << "Diana\n" << scoreB << ':' << scoreA << '\n';

}

**Задача G**

#include <iostream>

#include <vector>

#include <algorithm>

#include <limits>

using namespace std;

struct atm {long long x, c;}; const long long INF = 2e18;

long long calc(long long capacity, const long long w, const vector<atm>& v) {

vector<long long> dp(v.size(), INF);dp[0] = 0;

for (int i = 1; i < v.size(); ++i)

for (int j = 0; j < i; ++j)

if (w \* (v[i].x - v[j].x) <= capacity && dp[j] < INF)

dp[i] = min(dp[i], dp[j] + w \* (v[i].x - v[j].x) \* v[j].c);

return dp.back();

}

int main() { long long d, w, n; cin >> d >> w >> n;

vector<atm> v(n);

for (auto& atm : v) cin >> atm.c;

for (auto& atm : v) cin >> atm.x;

sort(v.begin(), v.end(), [](const atm& a, const atm& b) {

if (a.x == b.x) return a.c < b.c;

return a.x < b.x;});

v.erase(unique(v.begin(), v.end(), [](const atm& a, const atm& b) {

return a.x == b.x;}), v.end());

v.push\_back({ d, 0 });

long long minPrice = calc(numeric\_limits<long long>::max(), w, v),left = 0, right = 1e13;

while (left + 1 < right) { long long mid = (left + right) / 2;

if (calc(mid, w, v) <= minPrice) right = mid;else left = mid;}

cout << right;

}

**Задача H**

#include <iostream>

#include <vector>

#include <algorithm>

#include <string>

#define N 8

#define for\_k for(int k = 1; k < N; k++)

#define for\_0 for(int k = 1; k <= N; k++)

using namespace std;

const char queen = 'Q', king = 'K', pawn = 'p', rook = 'r', knight = 'n', bishop = 'b';

char a[N][N]; bool can[N][N];

bool isC(int i, int j) { return 0 <= i && i < 8 && 0 <= j && j < 8;}

string toC(int i, int j) { return string(1, 'a' + j) + to\_string(8 - i); }

int bI, bJ, kI, kJ; vector<string> f\_T\_B;

void ban(int i, int j) { if (isC(i, j)) {

if (i == kI && j == kJ) f\_T\_B.push\_back(toC(bI, bJ)); can[i][j] = false; } }

bool isE(int i, int j) { if (!isC(i, j)) return true; return a[i][j] == '.'; }

int main() {

ios\_base::sync\_with\_stdio(false);

cin.tie(nullptr);

for (int i = 0; i < N; ++i) { for (int j = 0; j < N; ++j) { cin >> a[i][j];

if (a[i][j] == king) { a[i][j] = '.'; kI = i; kJ = j; }

if (a[i][j] == '.') can[i][j] = true; else can[i][j] = false; } }

for (int i = -1; i <= 1; ++i) { for (int j = -1; j <= 1; ++j) {

if (i == 0 && j == 0) continue; if (isC(kI + i, kJ + j)) { can[kI + i][kJ + j] = true; } } }

for (int i = 0; i < N; ++i) { for (int j = 0; j < N; ++j) { bI = i; bJ = j;

if (a[i][j] == pawn) { ban(i - 1, j - 1); ban(i - 1, j + 1); }

else if (a[i][j] == rook) { for\_k { ban(i, j + k); if (!isE(i, j + k)) break; }

for\_k { ban(i, j - k); if (!isE(i, j - k)) break; }

for\_k { ban(i + k, j); if (!isE(i + k, j)) break; }

for\_k { ban(i - k, j); if (!isE(i - k, j)) break; }

} else if (a[i][j] == knight) { ban(i - 2, j - 1); ban(i - 2, j + 1);

ban(i - 1, j + 2); ban(i + 1, j + 2); ban(i + 2, j + 1);

ban(i + 2, j - 1); ban(i + 1, j - 2); ban(i - 1, j - 2);

} else if (a[i][j] == bishop) {

for\_0 { ban(i - k, j - k); if (!isE(i - k, j - k)) break; }

for\_0 { ban(i + k, j + k); if (!isE(i + k, j + k)) break; }

for\_0 { ban(i - k, j + k); if (!isE(i - k, j + k)) break; }

for\_0 { ban(i + k, j - k); if (!isE(i + k, j - k)) break;}

} else if (a[i][j] == queen) { for\_k { ban(i, j + k); if (!isE(i, j + k)) break; }

for\_k { ban(i, j - k); if (!isE(i, j - k)) break; }

for\_k { ban(i + k, j); if (!isE(i + k, j)) break; }

for\_k { ban(i - k, j); if (!isE(i - k, j)) break; }

for\_0 { ban(i - k, j - k); if (!isE(i - k, j - k)) break; }

for\_0 { ban(i + k, j + k); if (!isE(i + k, j + k)) break; }

for\_0 { ban(i - k, j + k); if (!isE(i - k, j + k)) break; }

for\_0 { ban(i + k, j - k); if (!isE(i + k, j - k)) break; }

} } } vector<string> ava;

for (int i = -1; i <= 1; ++i) for (int j = -1; j <= 1; ++j) { if (i == 0 && j == 0) continue;

if (isC(kI + i, kJ + j) && can[kI + i][kJ + j]) ava.push\_back(toC(kI + i, kJ + j));}

sort(ava.begin(), ava.end()); sort(f\_T\_B.begin(), f\_T\_B.end());

if (!can[kI][kJ] && ava.empty()) { cout << "Checkmate\n"; for (auto& i : f\_T\_B) cout << i << '\n'; }

else if (can[kI][kJ] && ava.empty()) cout << "Stalemate\n";

else { cout << "Continue\n"; for (auto& i : ava) cout << i << '\n'; }

}

