Bài 1

#include <iostream>

class Calculator {

private:

int a;

int b;

public:

Calculator() {

a = 0;

b = 0;

}

void inputNumbers() {

std::cin >> a >> b;

}

int calculateSum() {

return a + b;

}

void displayResult(int result) {

std::cout << result << std::endl;

}

};

int main() {

Calculator calculator;

calculator.inputNumbers();

int result = calculator.calculateSum();

calculator.displayResult(result);

return 0;

}

Bài 3

#include <iostream>

#include <vector>

#include <algorithm>

#include <climits>

using namespace std;

class Query {

private:

int u, v;

public:

Query(int uu, int vv) : u(uu), v(vv) {}

int getU() const {

return u;

}

int getV() const {

return v;

}

};

class Update {

private:

int u, v, k;

public:

Update(int uu, int vv, int kk) : u(uu), v(vv), k(kk) {}

int getU() const {

return u;

}

int getV() const {

return v;

}

int getK() const {

return k;

}

};

class ArrayUpdater {

private:

vector<int> arr;

public:

ArrayUpdater(int n) : arr(n, 0) {}

void applyUpdate(const Update& update) {

arr[update.getU() - 1] += update.getK();

if (update.getV() < arr.size()) {

arr[update.getV()] -= update.getK();

}

}

void calculatePrefixSum() {

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

arr[i] += arr[i - 1];

}

}

int getMaxValueInRange(int u, int v) const {

int maxVal = INT\_MIN;

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

maxVal = max(maxVal, arr[i]);

}

return maxVal;

}

};

int main() {

int n, m;

cin >> n >> m;

ArrayUpdater arrayUpdater(n);

for (int i = 0; i < m; ++i) {

int u, v, k;

cin >> u >> v >> k;

Update update(u, v, k);

arrayUpdater.applyUpdate(update);

}

arrayUpdater.calculatePrefixSum();

int p;

cin >> p;

for (int i = 0; i < p; ++i) {

int u, v;

cin >> u >> v;

Query query(u, v);

cout << arrayUpdater.getMaxValueInRange(query.getU(), query.getV()) << endl;

}

return 0;

}

