#include <iostream>

#include <omp.h>

#include <vector>

#include <sstream>

#include <string>

#include <algorithm>

#include <ctime>

using namespace std;

constexpr size\_t MIN\_LEN = 4;

constexpr size\_t MAX\_LEN = 8;

constexpr size\_t MAX\_VALUE = 100;

/// <summary>

/// Генерирует вектор строк

/// </summary>

/// <returns>вектор строк</returns>

vector<string> GenerateVector()

{

vector<string> v;

v.reserve(MIN\_LEN + (rand() % (MAX\_LEN - MIN\_LEN)));

for (size\_t i = 0; i < v.capacity(); i++)

{

stringstream ss;

ss << rand() % MAX\_VALUE;

v.push\_back(ss.str());

}

return v;

}

/// <summary>

/// Считывает строку представляющую множество из консоли

/// </summary>

/// <returns>Считанное множество</returns>

vector<string> InputVector()

{

vector<string> v;

string str;

getline(cin, str, '\n');

stringstream ss(str);

while (ss >> str)

{

v.push\_back(str);

}

return v;

}

/// <summary>

/// Выводит вектор в поток

/// </summary>

/// <typeparam name="T"></typeparam>

/// <param name="os">выходной поток</param>

/// <param name="v">вектор</param>

/// <returns></returns>

template<typename T>

ostream& operator<<(ostream& os, const vector<T>& v)

{

for (const auto& i : v)

{

os << "(" << i << ") ";

}

return os;

}

/// <summary>

/// параллельно вычисляет прямое произведение для A и B

/// </summary>

/// <param name="A">первое множество</param>

/// <param name="B">второе множество</param>

/// <returns>прямое произведение A и B</returns>

vector<string> Multiply(const vector<string>& A, const vector<string>& B)

{

vector<string> result(A.size() \* B.size());

#pragma omp parallel for ordered

for (size\_t i = 0; i < A.size(); i++)

{

//cout << omp\_get\_thread\_num() << "\n";

for (size\_t j = 0; j < B.size(); j++)

{

result[i \* B.size() + j] = A[i] + ", " + B[j];

}

}

return result;

}

int main()

{

srand(time(0));

// Генерируем массивы

//vector<string> A(move(GenerateVector()));

//vector<string> B(move(GenerateVector()));

//vector<string> C(move(GenerateVector()));

//vector<string> D(move(GenerateVector()));

// Выводим сгенерированные массивы массивы

//cout << "A = " << A << '\n';

//cout << "A = " << B << '\n';

//cout << "A = " << C << '\n';

//cout << "A = " << D << '\n';

cout << "[A] = ";

//cin.ignore(256, '\n');

vector<string> A = InputVector();

cout << "[B] = ";

vector<string> B = InputVector();

cout << "[C] = ";

vector<string> C = InputVector();

cout << "[D] = ";

vector<string> D = InputVector();

auto ab = Multiply(A, B);

cout << '\n' << "A \* B = " << '\n' << ab << '\n';

auto abc = Multiply(ab, C);

cout << '\n' << "(A\*B)\*C = " << '\n' << abc << '\n';

auto abcd = Multiply(abc, D);

cout << '\n' << "((A\*B)\*C)\*D = " << '\n' << abcd << '\n';

}