

Правительство Российской Федерации

**Федеральное государственное автономное образовательное
учреждение высшего образования «Национальный исследовательский
университет «Высшая школа экономики»**

Факультет компьютерных наук
Департамент программной инженерии

**Отчет к домашнему заданию
По дисциплине
«Архитектура вычислительных систем»**

Работу выполнил:
Студент группы БПИ-194 Горбач И.В.
Вариант 7

Москва 2020

Задание

7. Вычислить прямое произведение множеств $A_1, A_2, A_3 \dots A_n$. Входные данные: целое положительное число n , множества чисел $A_1, A_2, A_3 \dots A_n$, мощности множеств равны между собой и мощность каждого множества больше или равна 1. Количество потоков является входным параметром.

Модель

Итеративный параллелизм используется для реализации нескольких потоков (часто идентичных), каждый из которых содержит циклы. Потоки программы, описываются итеративными функциями и работают совместно над решением одной задачи.

Решение

Для реализации данной задачи была использована библиотека OpenMP

Ввод:

Сначала на вход подается 1 число- количество используемых множеств

Затем подается второе число- мощность множеств

```
std::cout << "Введите количество множеств: ";
std::cin >> n;
std::cout << "Введите количество чисел во множестве (n одинаково для всех множеств): ";
std::cin >> countUnit;
for(int i = 0; i < n; i++) {
    for(int j = 0; j < countUnit; j++) {
        int elementSet;
        std::cin >> elementSet;
        tempSet.insert(elementSet);
    }
    v.push_back(tempSet);
    tempSet.clear();
}
```

Затем идет работа со множествами, так как довольно трудно искать декартово произведение всех множеств, вывод множеств производился по вертикале в отдельные файлы с примерами, в связи с тем, что консоль предоставляет ограниченную длину строки, что ломает вывод при больших числах

Все примеры лежат в папке папке проекта в папке cmake-build-debug\Answer

Для распараллеливания были использованы директивы `#pragma omp parallel for`

Код программы

```
#include <iostream>
#include <vector>
#include <set>
#include <cmath>
#include "omp.h"
#include "fstream"

int main() {
    system("chcp 65001");
    std::ofstream outf("Answer/answer1.txt");
    int n = 0;
    int countUnit = 0;
    std::set<int> tempSet;
    std::set<int> :: iterator it;
    std::vector<std::set<int>> v;

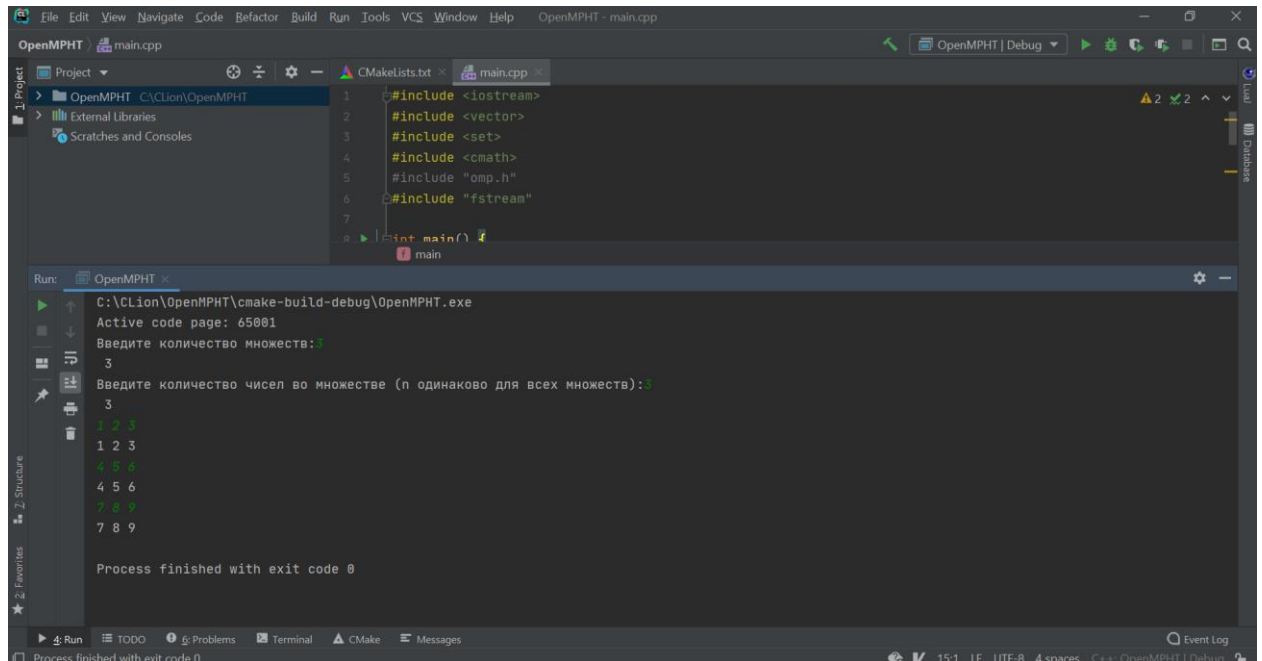
    std::cout << "Введите количество множеств: ";
    std::cin >> n;
    std::cout << "Введите количество чисел во множестве (n одинаково для всех\nмножеств): ";
    std::cin >> countUnit;
    for(int i = 0; i < n; i++) {
        for(int j = 0; j < countUnit; j++) {
            int elementSet;
            std::cin >> elementSet;
            tempSet.insert(elementSet);
        }
        v.push_back(tempSet);
        tempSet.clear();
    }

    int temp = pow(countUnit, n) / countUnit;
    int theRepeat = 0;
    int temp_i = 0;

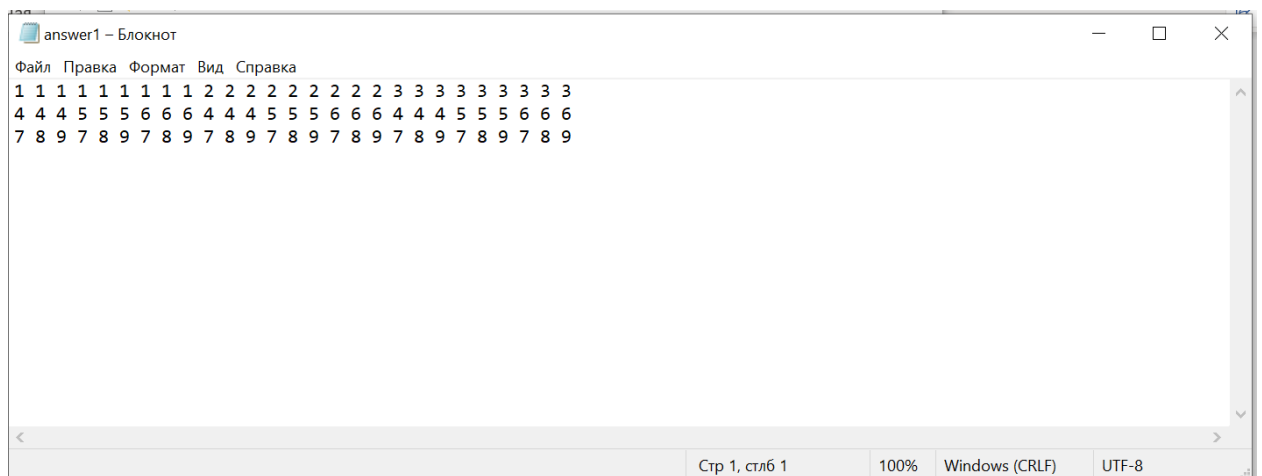
#pragma omp parallel for private(it)
    for(int i = 0; i < n; i++) {
        it = v[i].begin();
        std::vector<int> tempV;
#pragma omp parallel for private(tempV)
        for(int k = 0; k < countUnit; k++) {
            tempV.push_back(*it);
            it++;
        }
        for(int j = 0; j < pow(countUnit, n); j++) {
            if(theRepeat == temp) {
                theRepeat = 0;
                temp_i++;
                if(temp_i == countUnit)
                    temp_i = 0;
            }
            outf << tempV[temp_i]<< " ";
            theRepeat++;
        }
        outf << std::endl;
        temp /= countUnit;
        theRepeat = 0;
        temp_i = 0;
    }
    return 0;
}
```

Тестирование

Ввод:



Вывод:



Ввод 2:

