

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Департамент программной инженерии

**ПРОГРАММА, ВЫЧИСЛЯЮЩАЯ АЛГЕБРАИЧЕСКИЕ
ДОПОЛНЕНИЯ ДЛЯ КАЖДОГО ЭЛЕМЕНТА МАТРИЦЫ**

Задание 3

Вариант 3

Отчет

Исполнитель:

студентка группы БПИ196

_____ / А. А. Баранова /

« ____ » _____ 2020 г.

1.	ОПИСАНИЕ ЗАДАЧИ	3
2.	МОДЕЛЬ ВЫЧИСЛЕНИЙ	4
3.	ОПИСАНИЕ ФОРМАТА ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ	5
3.1.	Запуск программы.	5
3.2.	Описание входных данных.....	5
3.1.	Формат выходных данных.....	5
4.	ТЕСТОВЫЕ ПРИМЕРЫ	6
5.	СПИСОК ЛИТЕРАТУРЫ	7

1. ОПИСАНИЕ ЗАДАЧИ

1.1. Задание:

Найти алгебраическое дополнение для каждого элемента произвольной матрицы A размерности $n \times n$, где n – целое положительное число. Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

1.2. Разрабатываемая программа должна:

- 1) получать от пользователя пути к файлам с входными и выходными данными [см. п. 3.1];
- 2) обрабатывать информацию из входного файла, определяя необходимые для работы данные: количество потоков и исходную матрицу;
- 3) реализовывать алгоритм нахождения алгебраических дополнений элементов матрицы;
- 4) выполнять вычисление алгебраических дополнений элементов матрицы в разных потоках (количество потоков задается пользователем), используя итеративный параллелизм;
- 5) записывать в выходной файл полученные результаты [см. п. 3.2].

2. МОДЕЛЬ ВЫЧИСЛЕНИЙ

Для работы с потоками программа использует итеративный параллелизм. Каждый из реализуемых потоков идентичен и содержит циклы, в которых заполняется матрица алгебраических дополнений элементов, на которые отведен данный поток. Потоки программы, описываются итеративными функциями и работают совместно над решением задачи.

Каждый из m потоков в цикле находит алгебраические дополнения элементов $i, i+m, \dots$ (где i – индекс потока) матрицы $n \times n$ и записывает найденное значение в соответствующую ячейку результирующей матрицы алгебраических дополнений.

Итеративный параллелизм, как модель построения многопоточных программ, подробнее описан в работе [https://l.wzm.me/_coder/custom/parallel.programming/001.htm], а также в статье [https://en.wikipedia.org/wiki/Loop-level_parallelism].

3. ОПИСАНИЕ ФОРМАТА ВХОДНЫХ И ВЫХОДНЫХ ДАННЫХ

3.1. Запуск программы.

Для запуска приложения используется командная строка, аргументами служат две строки, записанные через пробел, – путь к файлу с входными данными и путь к файлу для записи выходных данных.

Из директории, содержащей task03.cpp, запустить программу можно, используя команды:

```
co task03.cpp  
task03 <input_path.txt> <output_path.txt>,
```

где на месте <input_path.txt> и <output_path.txt> должны соответственно быть указаны пути к файлам с входными и выходными данными.

3.2. Описание входных данных.

Для работы программы пользователь предоставляет приложению путь к текстовому файлу с информацией о количестве потоков и матрицей, с которой программа должна работать.

Требования к файлу с входными данными:

- 1) расширение .txt;
- 2) в первой строке файла записано единственное целое число m ($m > 0$) – количество потоков, которые необходимо использовать в работе программы;
- 3) во второй строке файла записано единственное целое число n ($n > 1$) – количество строк и столбцов квадратной матрицы;
- 4) на следующих n строках находятся по n целых чисел – элементы матрицы.

Матрица, записанная в файле, может иметь больший размер, чем $n \times n$, лишние данные будут проигнорированы. Элементы матрицы могут быть записаны как числа с плавающей точкой, в этом случае значения будут округлены вниз.

3.1. Формат выходных данных.

В файл, путь к которому был передан вторым аргументом, программа записывает матрицу $n \times n$ – матрицу алгебраических дополнений элементов исходной матрицы.

4. **ТЕСТОВЫЕ ПРИМЕРЫ**

Примеры входных и выходных данных отражены в прилагаемых файлах, расположенных в папках input (с файлами входных данных) и output (с файлами выходных данных). В папке input содержится 10 файлов с названиями формата testXX.txt, где на месте XX стоит номер теста (от 01 до 10). В папке output содержится 10 файлов с названиями формата answerXX.txt, где на месте XX стоит номер теста, соответствующего данному ответу (от 01 до 10). Каждый файл из папки output содержит результат работы программы при входных данных из соответствующего файла папки input.

Например, входные данные для теста №01 лежат в файле input\test01.txt, результат работы программы для теста №01 – в файле output\answer01.txt

5. СПИСОК ЛИТЕРАТУРЫ

- 1) Практические приемы построения многопоточных приложений: [Электронный ресурс]: Режим доступа: URL: <http://softcraft.ru/edu/comparch/tasks/t03/>, свободный. (дата обращения: 17.11.2020).
- 2) Loop-level parallelism: [Электронный ресурс]: Режим доступа: URL: https://en.wikipedia.org/wiki/Loop-level_parallelism, свободный. (дата обращения: 17.11.2020).
- 3) Алгебраическое дополнение: [Электронный ресурс]: Режим доступа: URL: https://ru.wikipedia.org/wiki/Алгебраическое_дополнение, свободный. (дата обращения: 17.11.2020).
- 4) Gregory R. Andrews. Foundations of Multithreaded, Parallel, and Distributed Programming. Режим доступа: URL: https://l.wzm.me/_coder/custom/parallel.programming/main.htm, свободный. (дата обращения: 17.11.2020) – University of Arizona.