МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ

(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Кафедра вычислительной математики и программирования

**спецкурс «Параллельные и распределенные вычисления»**

**ОТЧЕТ**

**Лабораторная работа № 2**

**«Программирование CUDA»**

Выполнил: Савельев А.С.

Группа: М8О-107-22

Преподаватель: Семенов С. А.

Москва, 2022

Содержание

[1. Постановка задачи 2](#_Toc7492)

[2. Описание решения 2](#_Toc16227)

[4. Основные моменты кода 3](#_Toc5215)

[5. Результат работы программы 4](#_Toc10579)

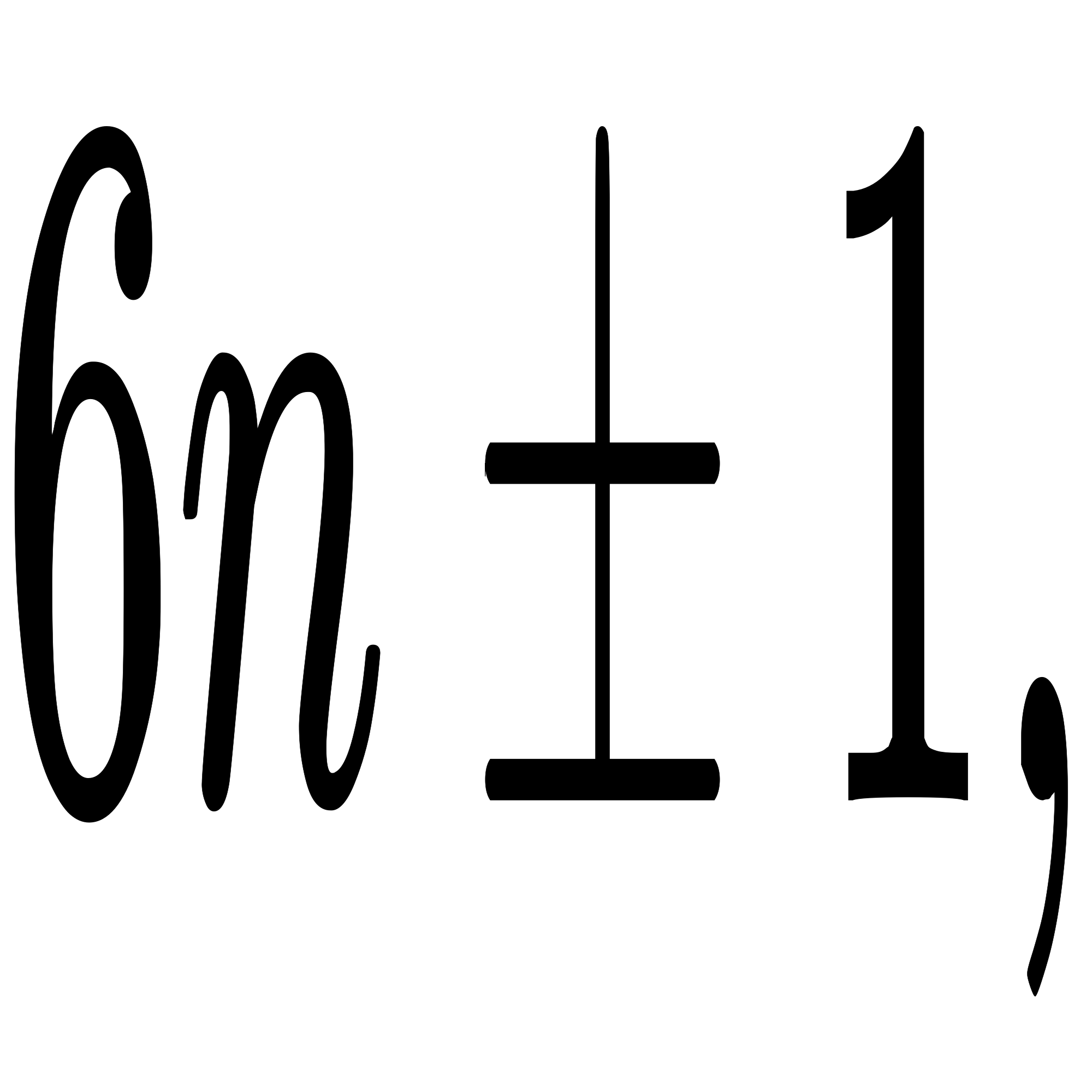
[7. Выводы 5](#_Toc8193)

[8. Приложения 5](#_Toc23729)

# 1. Постановка задачи

Вариант 12 (Лёгкий): Дано натуральное число N. Найти все меньшие n простые числа-близнецы, т.е. между которыми находится только одно составное число.

# Описание решения

Все пары чисел-близнецов, кроме (3, 5), имеют вид  так как числа с другими вычетами [по модулю](https://ru.wikipedia.org/wiki/%D0%A1%D1%80%D0%B0%D0%B2%D0%BD%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D0%BE_%D0%BC%D0%BE%D0%B4%D1%83%D0%BB%D1%8E" \o "Сравнение по модулю) 6 делятся на 2 или на 3. Таким образом решение сводится к параллельному заполнению массива числами по заданному правилу.

Результатами выполнения программы являются значения массива *pairs*, полученных на GPU.

# Основные моменты кода

Генерация чисел по заданному правилу:

\_\_global\_\_ void twinPrime(int \* pairs)

{

\_\_shared\_\_ int cache[threadsPerBlock];

int tid = threadIdx.x + blockIdx.x \* blockDim.x;

cache[threadIdx.x] = tid / 2 \* 6 + (tid % 2 ? 1 : -1);

pairs[tid] = cache[threadIdx.x];

}

Дополнение числами-исключениями:

// Передача данных хосту

cudaMemcpy(pairs, dev\_pairs, (pairsCount \* 2 + 2) \* sizeof(int), cudaMemcpyDeviceToHost);

pairs[0] = 3;

pairs[1] = 5;

# 5. Результат работы программы

# 

1. Выводы

В Лабораторной работе №2 были изучены возможности распараллеливания и разделяемой памяти в контексте задачи по поиску чисел-близнецов. Рост производительности GPU относительно CPU коррелирует с ростом размера обрабатываемого массива значений.

# Приложения

https://github.com/Ansave/M806\_XR\_AI/tree/main/CUDA\_Homework/LR2\_E12/LR2\_E12