МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего профессионального образования
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА №14

РАБОТА ЗАЩИЩЕНА С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

старший преподаватель		А. Ю. Сыщиков
должность, уч. степень, звание	подпись, дата	инициалы, фамилия

ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 2
OpenMP, Распараллеливание вычислений

по дисциплине: Системы с параллельной обработкой информации

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. 1742 Д.В. Коробков

подпись, дата — инициалы, фамилия

Санкт-Петербург 2021

1. Цель работы

Реализовать и распараллелить с помощью OpenMP различные алгоритмы.

№ варианта	Размерность задачи	Тип элемента вектора	Алгоритм
9	90x60	Без знаковый целый	Сортировка столбцов матрицы по убыванию

2. Текст программы

```
#include <omp.h>
#include <time.h>
#include <locale.h>
#include <iostream>
using namespace std;
constexpr auto N = 5, M =5;
int main(int argc, char* argv[])
{
	setlocale(0, "");
	unsigned int A[N][M];
	for (int i = 0; i < N; i++) {
		for (int j = 0; j < M; j++) {
			A[i][j] = rand() % 100;
		}
	}
	cout << "A:";
	for (int i = 0; i < N; i++) {
		for (int j = 0; j < M; j++) {
			cout << " " << A[i][j];
		}
		cout << "\n  ";
	}
	cout << endl;
	int i, j,n,k;
#pragma omp parallel shared(A) private(i, n, j, k)
	{
		n = omp_get_thread_num();
		#pragma omp for
		for (j = 0; j < M; j++)
```

```
		{
			printf("Нить %d сортирует столбец %d\n", n, j);
			for (k = 0; k < N - 1; k++) {
				for (i = 0; i < N - 1 - k; i++) {
					if (A[i][j] < A[i + 1][j])
					{
						printf("\tНить %d сравнила элементы [%d][%d] и [%d][%d] и поменяла их местами\n", n, i, j, i + 1, j);
						unsigned int temp = A[i][j];
						A[i][j] = A[i + 1][j];
						A[i + 1][j] = temp;
					}
					else {
						printf("\t\tНить %d сравнивает элементы [%d][%d] и [%d][%d] и не поменяла их.\n", n, i, j, i + 1, j);
					}
				}
			}
		}
		cout << "A:";
		for (int i = 0; i < N; i++) {
			for (int j = 0; j < M; j++) {
				cout << " " << A[i][j];
			}
			cout << "\n  ";
		}
		cout << endl;
}
```

3. Результат работы программы

```
A: 41 67 34 0 69
   24 78 58 62 64
   5 45 81 27 61
   91 95 42 27 36
   91 4 2 53 92

Нить 3 сортирует столбец 3
        Нить 3 сравнила элементы [0][3] и [1][3] и поменяла их местами
Нить 4 сортирует столбец 4
                Нить 4 сравнивает элементы [0][4] и [1][4] и не поменяла их.
                Нить 4 сравнивает элементы [1][4] и [2][4] и не поменяла их.
                Нить 4 сравнивает элементы [2][4] и [3][4] и не поменяла их.
        Нить 4 сравнила элементы [3][4] и [4][4] и поменяла их местами
                Нить 4 сравнивает элементы [0][4] и [1][4] и не поменяла их.
                Нить 4 сравнивает элементы [1][4] и [2][4] и не поменяла их.
        Нить 4 сравнила элементы [2][4] и [3][4] и поменяла их местами
                Нить 4 сравнивает элементы [0][4] и [1][4] и не поменяла их.
        Нить 4 сравнила элементы [1][4] и [2][4] и поменяла их местами
        Нить 4 сравнила элементы [0][4] и [1][4] и поменяла их местами
Нить 1 сортирует столбец 1
        Нить 1 сравнила элементы [0][1] и [1][1] и поменяла их местами
                Нить 1 сравнивает элементы [1][1] и [2][1] и не поменяла их.
        Нить 1 сравнила элементы [2][1] и [3][1] и поменяла их местами
                Нить 1 сравнивает элементы [3][1] и [4][1] и не поменяла их.
                Нить 1 сравнивает элементы [0][1] и [1][1] и не поменяла их.
        Нить 1 сравнила элементы [1][1] и [2][1] и поменяла их местами
                Нить 1 сравнивает элементы [2][1] и [3][1] и не поменяла их.
        Нить 1 сравнила элементы [0][1] и [1][1] и поменяла их местами
                Нить 1 сравнивает элементы [1][1] и [2][1] и не поменяла их.
                Нить 1 сравнивает элементы [0][1] и [1][1] и не поменяла их.
Нить 2 сортирует столбец 2
        Нить 2 сравнила элементы [0][2] и [1][2] и поменяла их местами
        Нить 2 сравнила элементы [1][2] и [2][2] и поменяла их местами
        Нить 2 сравнила элементы [2][2] и [3][2] и поменяла их местами
                Нить 2 сравнивает элементы [3][2] и [4][2] и не поменяла их.
        Нить 2 сравнила элементы [0][2] и [1][2] и поменяла их местами
                Нить 2 сравнивает элементы [1][2] и [2][2] и не поменяла их.
                Нить 2 сравнивает элементы [2][2] и [3][2] и не поменяла их.
                Нить 2 сравнивает элементы [0][2] и [1][2] и не поменяла их.
                Нить 2 сравнивает элементы [1][2] и [2][2] и не поменяла их.
                Нить 2 сравнивает элементы [0][2] и [1][2] и не поменяла их.
Нить 0 сортирует столбец 0
                Нить 0 сравнивает элементы [0][0] и [1][0] и не поменяла их.
                Нить 0 сравнивает элементы [1][0] и [2][0] и не поменяла их.
        Нить 0 сравнила элементы [2][0] и [3][0] и поменяла их местами
        Нить 0 сравнила элементы [3][0] и [4][0] и поменяла их местами
                Нить 0 сравнивает элементы [0][0] и [1][0] и не поменяла их.
        Нить 0 сравнила элементы [1][0] и [2][0] и поменяла их местами
        Нить 0 сравнила элементы [2][0] и [3][0] и поменяла их местами
        Нить 0 сравнила элементы [0][0] и [1][0] и поменяла их местами
        Нить 0 сравнила элементы [1][0] и [2][0] и поменяла их местами
                Нить 0 сравнивает элементы [0][0] и [1][0] и не поменяла их.
        Нить 3 сравнила элементы [1][3] и [2][3] и поменяла их местами
        Нить 3 сравнила элементы [2][3] и [3][3] и поменяла их местами
        Нить 3 сравнила элементы [3][3] и [4][3] и поменяла их местами
                Нить 3 сравнивает элементы [0][3] и [1][3] и не поменяла их.
                Нить 3 сравнивает элементы [1][3] и [2][3] и не поменяла их.
        Нить 3 сравнила элементы [2][3] и [3][3] и поменяла их местами
                Нить 3 сравнивает элементы [0][3] и [1][3] и не поменяла их.
        Нить 3 сравнила элементы [1][3] и [2][3] и поменяла их местами
                Нить 3 сравнивает элементы [0][3] и [1][3] и не поменяла их.
A: 91 95 81 62 92
   91 78 58 53 69
   41 67 42 27 64
   24 45 34 27 61
   5 4 2 0 36
```

Рисунок 1. Результат работы программы.