

# Задание №4

---

## Условие задания (вариант №1)

Вычислить векторное произведение квадратных матриц **A** и **B**. Входные данные: произвольные квадратные матрицы **A** и **B** одинаковой размерности. Размер матриц задается входным параметром. Количество потоков является входным параметром, при этом размерность матриц может быть не кратна количеству потоков.

## Отчёт о работе

### Код программы

Программа выполнена в виде консольного приложения. Ввод осуществляется через консоль, в качестве аргументов командной строки передаются количество потоков и размер матриц.

Реализация основана на **итеративном параллелизме** с использованием **OpenMP**. Каждый поток отвечает за подсчет своей части матрицы, притом все эти части, кроме, иногда, последней, одинакового размера.

## Тестирование программы

### 1. Стандартный ввод

```
"/Users/Floydan/Documents/Programming_projects/C++/Multithread homework/Homework 1/cmake-build-debug/Multithread homework" 3 3
Введите матрицу A:
5 10 15
20 25 30
35 40 45
Введите матрицу B:
10 0 0
10 50 50
0 50 50
Закончился подсчет элементов отрезка [0,3)
Закончился подсчет элементов отрезка [3,6)
Закончился подсчет элементов отрезка [6,9)

A * B =
750 1250 1250
2250 2750 2750
3750 4250 4250

Process finished with exit code 0
```

## 2. Стандартный ввод, но количество потоков больше максимального

```
"/Users/Floydan/Documents/Programming_projects/C++/Multithread homework/HomeWork 1/cmake-build-debug/Multithread_homework" 10 3
Количество потоков превышает допустимое значение, равное 9, будет использоваться максимальное количество
Введите матрицу A:
1 2 3
4 3 6
7 8 9
Введите матрицу B:
2 0 0
0 2 0
0 0 2
Закончился подсчет элементов отрезка [0,1)
Закончился подсчет элементов отрезка [3,4)
Закончился подсчет элементов отрезка [2,3)
Закончился подсчет элементов отрезка [5,6)
Закончился подсчет элементов отрезка [6,7)
Закончился подсчет элементов отрезка [7,8)
Закончился подсчет элементов отрезка [1,2)
Закончился подсчет элементов отрезка [4,5)
Закончился подсчет элементов отрезка [8,9)

A * B =
2 4 6
8 10 12
14 16 18

Process finished with exit code 0
```

В данном случае программа будет работать с максимальным допустимым количеством потоков, который зависит от размера матрицы

## 3. Неверный ввод строки

```
"/Users/Floydan/Documents/Programming_projects/C++/Multithread homework/HomeWork 1/cmake-build-debug/Multithread_homework" 1 2
Введите матрицу A:
5
Строка 1 введена неправильно, попробуйте ещё раз
100 100
aaaaa
Строка 2 введена неправильно, попробуйте ещё раз
1001's string
Строка 2 введена неправильно, попробуйте ещё раз
100 100
Введите матрицу B:
500
Строка 1 введена неправильно, попробуйте ещё раз
500 500
abc
Строка 2 введена неправильно, попробуйте ещё раз
500 500
Закончился подсчет элементов отрезка [0,4)

A * B =
100000 100000
100000 100000

Process finished with exit code 0
```

Ввод строки матрицы происходит, пока она не будет содержать хотя бы 2 целых числа

## 4. Ввод больших случайных матриц 50x50

```

Закончился подсчет элементов отрезка [2350,2375)
Закончился подсчет элементов отрезка [2400,2425)
Закончился подсчет элементов отрезка [2425,2450)
Закончился подсчет элементов отрезка [2450,2475)
Закончился подсчет элементов отрезка [2475,2500)
Закончился подсчет элементов отрезка [900,925)
Закончился подсчет элементов отрезка [925,950)
Закончился подсчет элементов отрезка [950,975)
Закончился подсчет элементов отрезка [975,1000)
Закончился подсчет элементов отрезка [1025,1050)
Закончился подсчет элементов отрезка [1350,1375)
Закончился подсчет элементов отрезка [1400,1425)

A * B =
99478 111412 96758 86732 128208 96650 109434 118774 114738 90333 96734 102209 125325 113285 106915 8515
110783 142798 131252 118008 151974 125030 140994 145694 140500 104397 106589 126349 159202 154466 136792 1163
115244 136516 124717 125484 169650 114494 141296 147068 142533 115320 109836 128703 164094 136379 145094 1013
115604 142919 131691 117511 153576 118724 141307 152065 133830 120846 126324 123290 155685 144924 125398 1104
99495 115877 101959 111645 126924 100358 115886 120761 100104 92586 94384 109301 119295 129998 114801 1010
97985 114699 101577 100323 127527 100338 108278 138355 107183 89793 102315 111394 128045 113768 114623 8626
109639 134112 132426 107630 154482 108392 150884 146098 139514 106019 113590 121503 157503 142622 140301 1149
93947 134850 115856 100047 130685 101920 118568 145260 112003 98474 120369 109346 134658 125037 103921 9417
134049 163918 127761 140807 185692 131942 174343 167443 149131 124974 123851 147103 177865 162488 154951 1238
104203 127208 122205 111672 145205 106082 129726 137146 127312 94679 100456 125535 146634 130427 130160 9306
126914 158579 146433 144041 185271 128267 165604 165671 157708 131850 129934 159233 176946 164988 152269 1339
101542 121628 108779 103315 130203 88015 110206 132324 113242 99642 106173 122461 132854 114272 109087 8996
95798 116293 117606 102304 135208 101570 114253 141693 110823 96526 97040 114620 136492 124303 116746 9430
106656 131742 107809 102700 138807 99153 134220 132145 110292 108076 104850 117646 146271 129090 125759 9383
91542 125548 106756 98841 126454 101833 115985 124065 104091 88318 105247 101678 127200 137062 105833 8843
110381 134909 114598 103897 149248 105000 140020 142947 125810 113942 111940 122716 152190 135939 129961 1029
107209 115837 125551 99385 138819 107166 134843 143096 111236 98053 117088 124131 150090 143872 117621 1021
101507 110529 110932 101039 146238 98702 131646 130511 142401 105189 102328 110507 140951 133323 133603 9269
125953 154570 140081 113784 174534 120237 152426 169503 136082 130905 129513 152083 173768 156603 143239 1122
108640 130148 109649 110069 132962 106967 135119 125962 128252 105700 115722 109149 136822 133494 119435 1075
107008 127905 105571 105202 145308 109944 120175 135106 119235 99247 109468 128337 140009 135536 121324 9516
100094 134451 127240 107649 146207 106247 130638 130908 123543 112224 114027 110748 147532 140265 122962 9835
109181 133307 130388 113907 145160 115229 143457 151924 123675 104538 110454 122198 149119 148158 135496 1074
112415 133927 109849 118055 133551 119746 126556 144611 125578 119011 106188 121984 147955 143494 131246 1139
103752 122130 123928 119730 151153 106050 137002 149185 127129 106001 124135 128294 154694 145864 124281 9520
115836 139580 131485 124778 151542 114822 143658 153151 145175 121053 129378 129562 158345 144914 143473 1081
99080 109721 105545 103899 129260 103864 118770 111309 103609 91358 90519 109033 132038 120206 104407 8680
109689 130475 118279 119764 152019 107597 144448 144260 132379 107018 117280 130103 143327 145298 138649 1130
110864 133937 115334 108876 143561 114025 135904 140663 116067 114327 107863 121882 155779 130325 115285 1003
106915 133200 121186 107898 128780 109499 122687 141955 102000 89678 106127 114609 143811 132866 122846 1010

```

К сожалению, тест не умещается в скриншот, однако обрабатывается верно