**Программное обеспечение, использованное при выполнении:** Visual Studio 2017, IPS 2019, Windows 10 – 64x

**Процессор** – четырехъядерный Intel Core i5 8250U с частотой 1.6 ГГц

1) Разберите программу, представленную в файле task\_for\_lecture5.cpp. Предварительный просмотр документа. В программе создается 2 потока, каждый из которых вычисляет средние значения матрицы, один по строкам исходной матрицы matrix, а другой - по столбцам. Запустите программу и убедитесь в ее работоспособности.

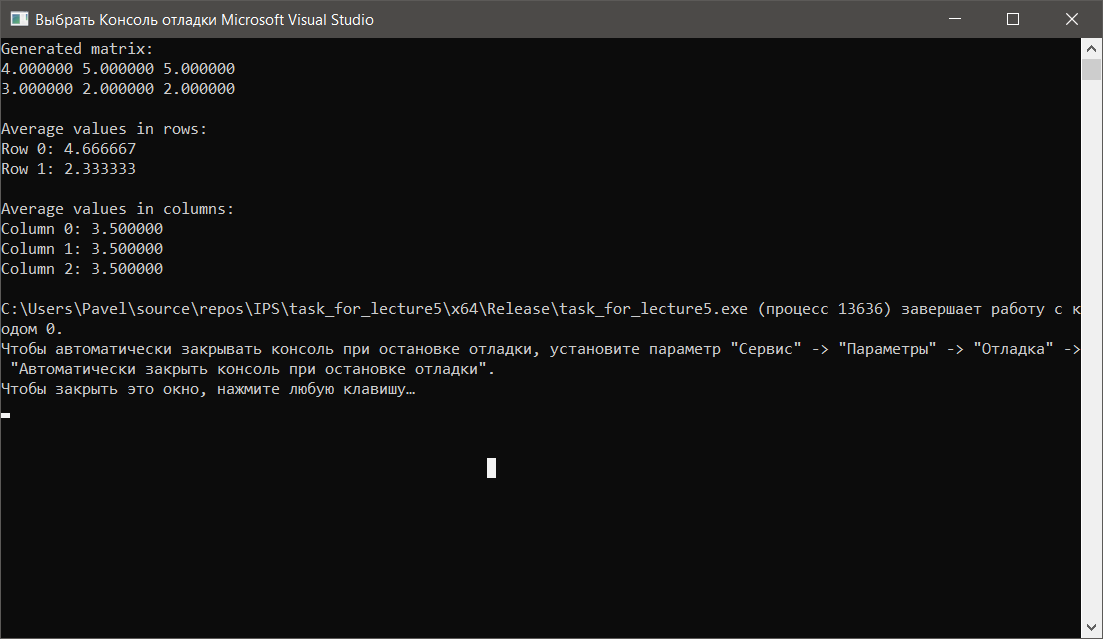


Рисунок 1 – результат работы программы при первом запуске

2) Проанализируйте программу и введите в нее изменения, которые по Вашему мнению повысят ее производительность.

Модифицируем программу, добавив циклы **cilk\_for** для распараллеливанияи необходимые **reducer** для суммирования:

|  |
| --- |
| case eprocess\_type::by\_rows:  {  cilk\_for ( size\_t i = 0; i < numb\_rows; ++i )  {  cilk::reducer<cilk::op\_add<double>> sum(0.0);  cilk\_for(size\_t j = 0; j < numb\_cols; ++j)  {  \*sum += matrix[i][j];  }  average\_vals[i] = sum.get\_value() / numb\_cols;  }  break;  }  case eprocess\_type::by\_cols:  {  cilk\_for ( size\_t j = 0; j < numb\_cols; ++j )  {  cilk::reducer<cilk::op\_add<double>> sum(0.0);  cilk\_for(size\_t i = 0; i < numb\_rows; ++i)  {  \*sum += matrix[i][j];  }  average\_vals[j] = sum.get\_value() / numb\_rows;  }  break;  } |

3) Определите с помощью ***Intel Parallel Inspector*** наличие в программе таких ошибок как: *взаимная блокировка*, *гонка данных*, *утечка памяти*. Сделайте скрины результатов анализа ***Parallel Inspector***(вкладки ***Summary***, ***Bottom-up***) для всех упомянутых ошибок*,*где отображаются обнаруженные ошибки, либо отражается их отсутствие. Запускайте анализы на разных уровнях (***Narrowest***, ***Medium***, ***Widest***).

- Запустим Intel Inspector на выявление взаимных блокировок и гонок данных

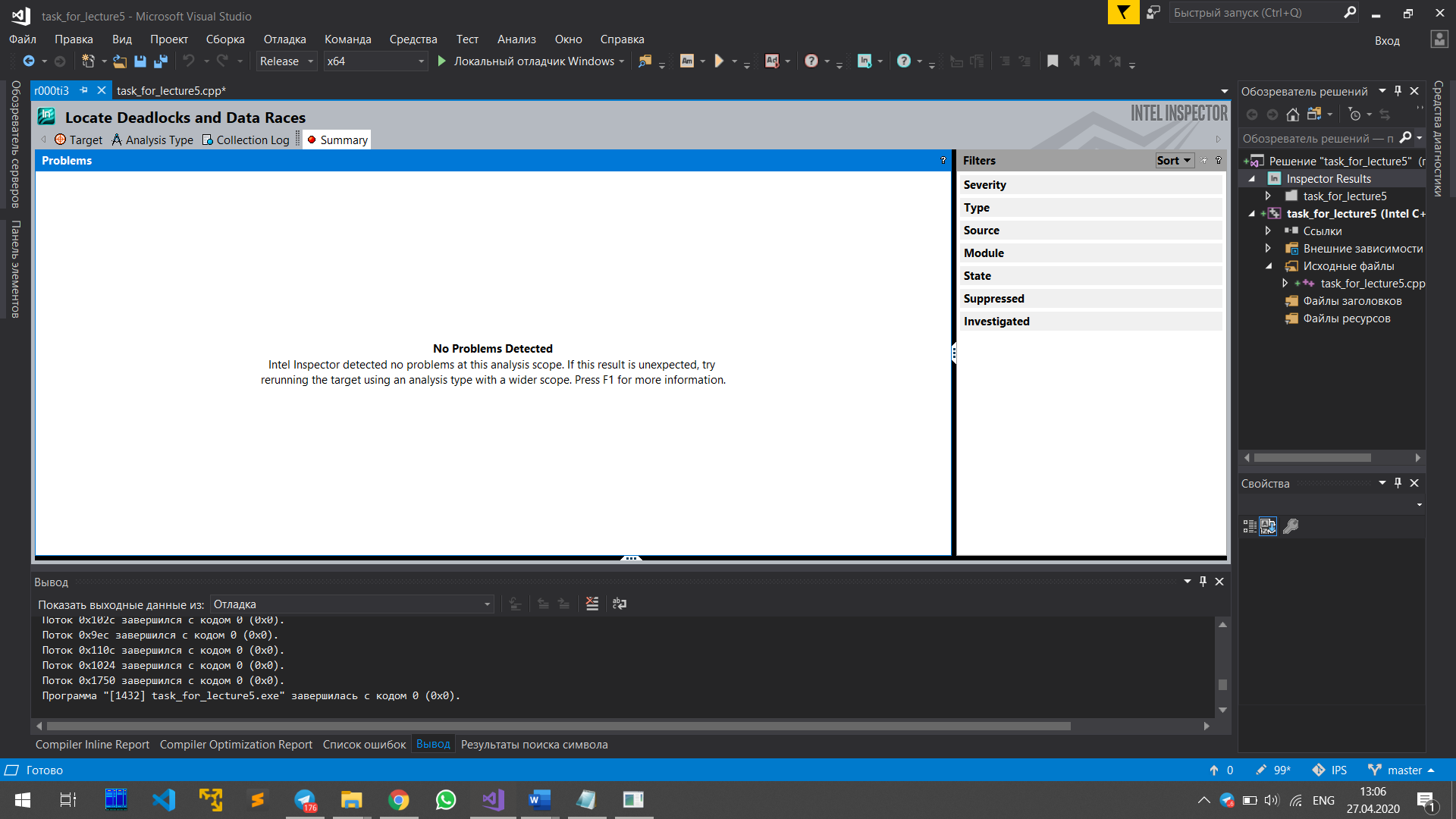


Рисунок 2 – результат работы Inspector на наличие гонок данных

Проблем не обнаружено

- Запустим Intel Inspector на выявление утечек данных

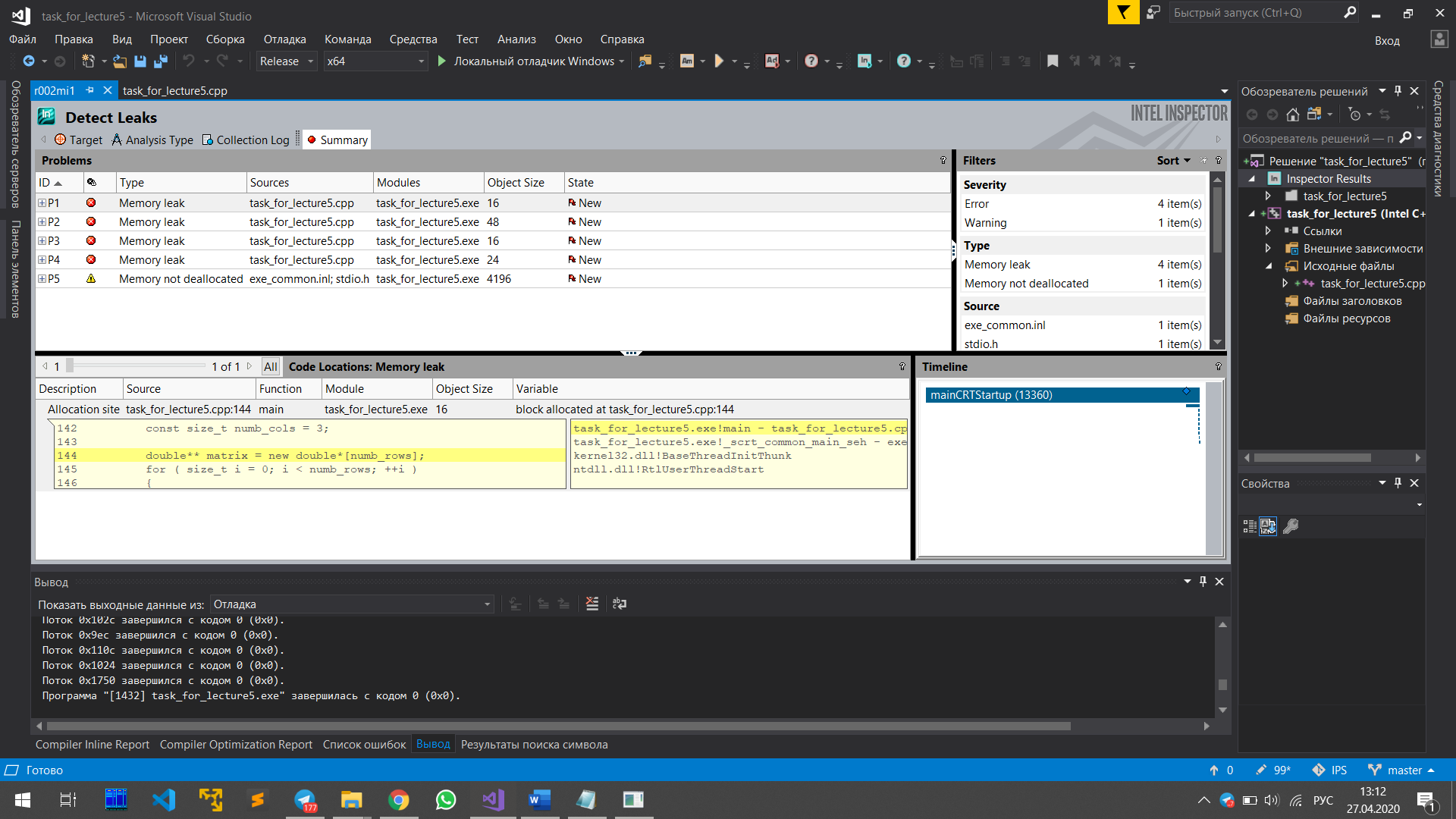


Рисунок 3 – результат работы Inspector на наличие утечек памяти

Как можно увидеть на скриншоте обнаружены утечки памяти, скорее всего разработчик забыл освободить выделенную под данные память. Исправим эту ошибку в следующем пункте.

4) Измените код программы таким образом, чтобы ***Inspector*** при проверке не находил в программе ошибок, перечисленных в **п. 3.**Сделайте скрины результатов запуска***Parallel Inspector***.

Внесем следующие изменения в конец программы:

|  |
| --- |
| for (size\_t i = 0; i < numb\_rows; ++i)  delete[]matrix[i];  delete[]matrix;  delete[]average\_vals\_in\_rows;  delete[]average\_vals\_in\_cols; |

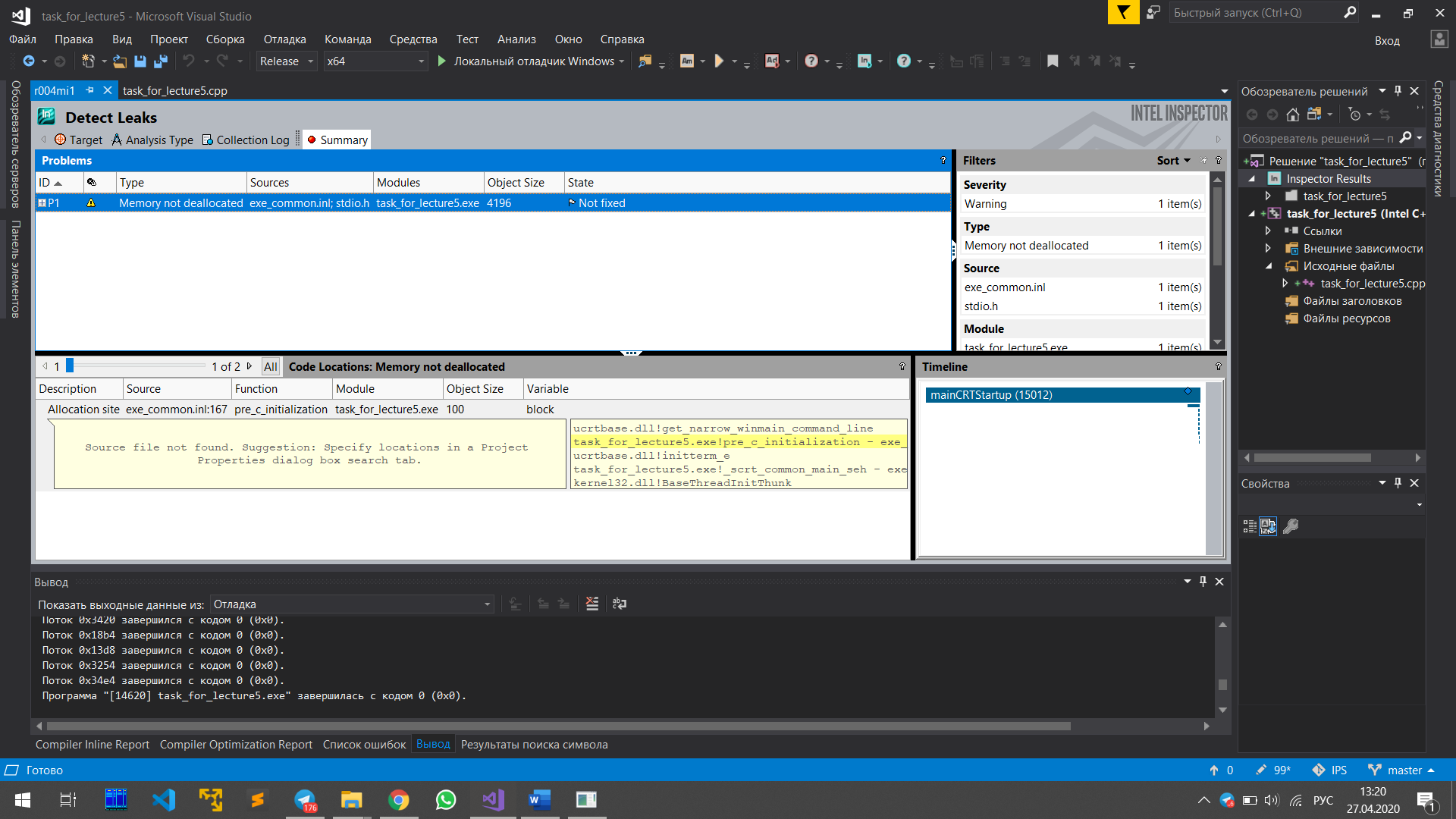


Рисунок 4 – результат работы Inspector на наличие утечек памяти

Как видно на скриншоте ошибки теперь отсутствуют.

**Вывод**: таким образом, в рамках данной лабораторной работы я проверил работоспособность программы, которая вычисляет средние значения матрицы по строкам и столбцам, убедился в ее работоспособности. Модифицировал программу, добавив **cilk\_for** и **reducer**, повысив ее производительность. Произвел анализ на предмет гонок данных и утечек памяти. Обнаружил и исправил ошибку, связанную с наличие утечек памяти.