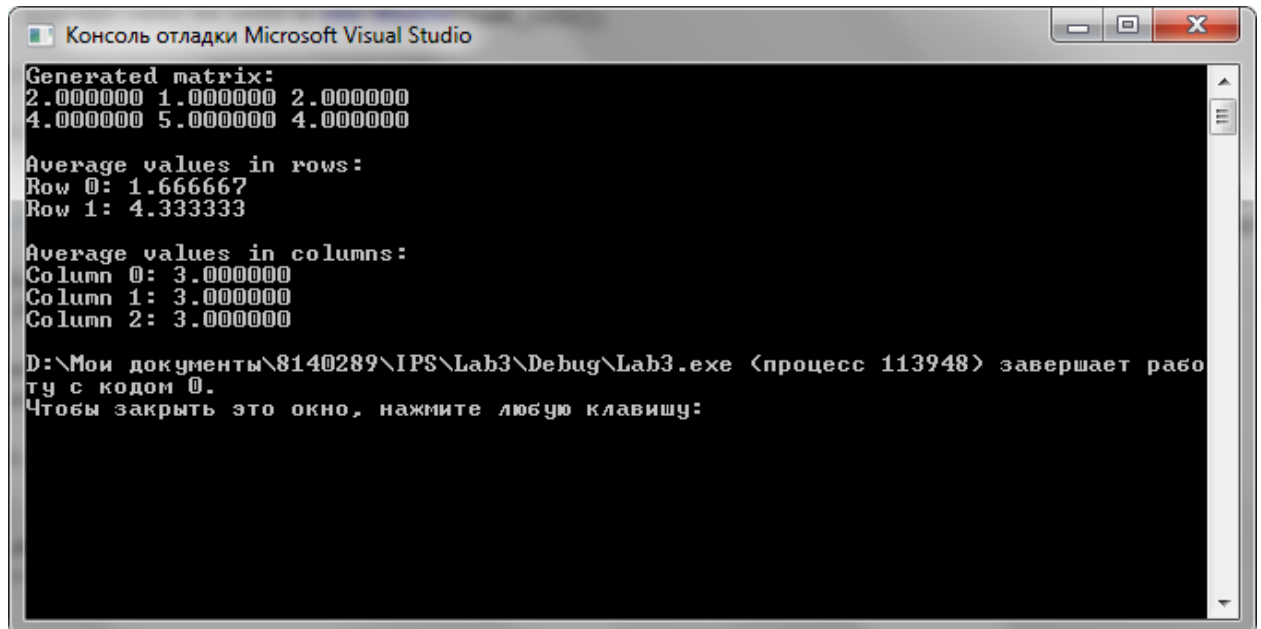


## Задание к занятию 5

1. Разберите программу представленную в файле [task\\_for\\_lecture5.cpp](#). 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.

*Запуск исходной программы:*



```
Консоль отладки Microsoft Visual Studio
Generated matrix:
2.000000 1.000000 2.000000
4.000000 5.000000 4.000000

Average values in rows:
Row 0: 1.666667
Row 1: 4.333333

Average values in columns:
Column 0: 3.000000
Column 1: 3.000000
Column 2: 3.000000

D:\Мои документы\8140289\IPS\Lab3\Debug\Lab3.exe (процесс 113948) завершает работу с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:
```

*Программа работает корректно.*

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

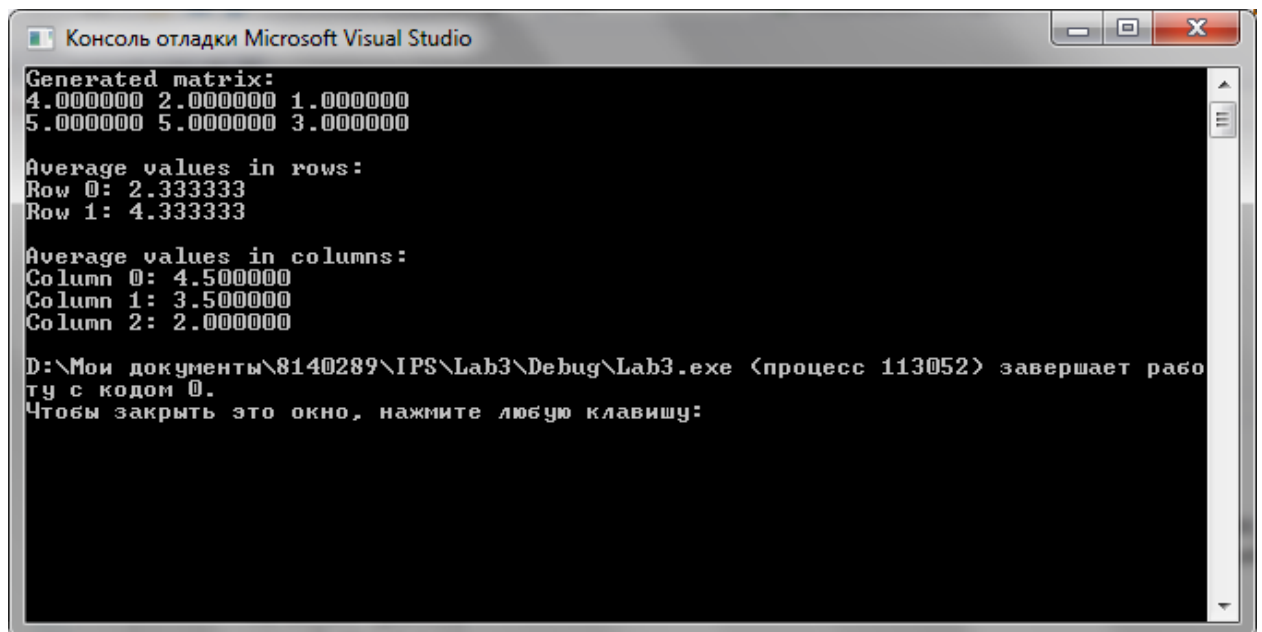
*Внесем изменения:*

```
void FindAverageValues( eprocess_type proc_type, double** matrix, const size_t
numb_rows, const size_t numb_cols, double* average_vals )
{
    switch ( proc_type )
    {
        case eprocess_type::by_rows:
        {
            cilk_for ( size_t i = 0; i < numb_rows; ++i )
            {
                //double sum( 0.0 );
                cilk::reducer_opadd<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 )
        {
            //double sum( 0.0 );
            cilk::reducer_opadd<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;
    }
    default:
    {
        throw("Incorrect value for parameter 'proc_type' in function
FindAverageValues() call!");
    }
}

```



Консоль отладки Microsoft Visual Studio

```

Generated matrix:
4.000000 2.000000 1.000000
5.000000 5.000000 3.000000

Average values in rows:
Row 0: 2.333333
Row 1: 4.333333

Average values in columns:
Column 0: 4.500000
Column 1: 3.500000
Column 2: 2.000000

D:\Мои документы\8140289\IPS\Lab3\Debug\Lab3.exe (процесс 113052) завершает рабо
ту с кодом 0.
Чтобы закрыть это окно, нажмите любую клавишу:

```

После изменений программа также работает правильно.

- 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**).

Определим проблемы, связанные с памятью. Запустим **Intel Parallel Inspector** в трех режимах (*Detect Leaks, Detect memory problems, Locate memory problems*).

Detect Leaks

Target

Analysis Type

Collection Log

Summary

Problems

ID	Type	Sources	Modules	Object Size	State
P1	Mismatched allocation/deallocation	[Unknown]	ucrtbase.dll; vcruntime140.dll		New
P2	Memory leak	task_for_lecture5.cpp	lab3.exe	8	New
P3	Memory leak	task_for_lecture5.cpp	lab3.exe	48	New
P4	Memory leak	task_for_lecture5.cpp	lab3.exe	16	New
P5	Memory leak	task_for_lecture5.cpp	lab3.exe	24	New
P6	Memory not deallocated	stdio.h	lab3.exe	4096	New

Filters

Severity

Error5 item(s)

Warning1 item(s)

Type

Memory leak4 item(s)

Memory not deallocated1 item(s)

Mismatched allocation/dealloc...1 item(s)

Source

[Unknown]1 item(s)

stdio.h1 item(s)

task\_for\_lecture5.cpp4 item(s)

1 of 1

All

Code Locations: Mismatched allocation/deallocation

Description	Source	Function	Module	Object Size	Variable
Mismatched deallocation site	vcruntime140.dll:0x78b2	set_se_translator	vcruntime140.dll	block	
Symbol information not found. Suggestion: Specify locations in a Project Properties dialog box search tab, then re-resolve the result.					
Allocation site	ucrtbase.dll:0x2f9c6	calloc_base	ucrtbase.dll	block	
Symbol information not found. Suggestion: Specify locations in a Project Properties dialog box search tab, then re-resolve the result.					

vcruntime140.dll!set\_se\_translator

ntdll.dll!RtlIsCurrentThreadAttachExempt

ntdll.dll!LdrShutdownThread

ntdll.dll!RtlExitUserThread

kernelbase.dll!FreeLibraryAndExitThread

ucrtbase.dll!calloc\_base

vcruntime140.dll!set\_se\_translator

ntdll.dll!RtlQueryEnvironmentVariable

ntdll.dll!RtlAllocateActivationContextSta

ntdll.dll!RtlDecodePointer

Timeline

o\_Clpow (120668)

Locate Memory Problems

Target

Analysis Type

Collection Log

Summary

Problems

ID	Type	Sources	Modules	Object Size	State
P1	Memory leak	task_for_lecture5.cpp	lab3.exe	8	New
P2	Memory leak	task_for_lecture5.cpp	lab3.exe	48	New
P3	Memory leak	task_for_lecture5.cpp	lab3.exe	16	New
P4	Memory leak	task_for_lecture5.cpp	lab3.exe	24	New
P5	Memory not deallocated	stdio.h	lab3.exe	4096	New

Filters

Severity

Error4 item(s)

Warning1 item(s)

Type

Memory leak4 item(s)

Memory not deallocated1 item(s)

Source

stdio.h1 item(s)

task\_for\_lecture5.cpp4 item(s)

Module

lab3.exe5 item(s)

1 of 1

All

Code Locations: Memory leak

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	task_for_lecture5.cpp:141	main	lab3.exe	8		block allocated at task_for_lecture5.cpp:141
139 const size_t numb_cols = 3;						
140						
141 double** matrix = new double*[numb_rows];						
142 for ( size_t i = 0; i < numb_rows; ++i )						
143 {						

lab3.exe!main - task\_for\_lecture5.cpp:141

lab3.exe!\_scrt\_common\_main\_seh - exe\_commo

kernel32.dll!BaseThreadInitThunk

ntdll.dll!RtlInitializeExceptionChain

ntdll.dll!RtlInitializeExceptionChain

Timeline

RtlQueryEnvironmentVariable (119872)

**Intel Inspector** Detect Memory Problems

Target Analysis Type Collection Log Summary

ID	Type	Sources	Modules	Object Size	State
P1	Memory leak	task_for_lecture5.cpp	lab3.exe	8	New
P2	Memory leak	task_for_lecture5.cpp	lab3.exe	48	New
P3	Memory leak	task_for_lecture5.cpp	lab3.exe	16	New
P4	Memory leak	task_for_lecture5.cpp	lab3.exe	24	New
P5	Memory not deallocated	stdio.h	lab3.exe	4096	New

**Filters**

Severity: Error (4 item(s)), Warning (1 item(s))

Type: Memory leak (4 item(s)), Memory not deallocated (1 item(s))

Source: stdio.h (1 item(s)), task\_for\_lecture5.cpp (4 item(s))

Module: lab3.exe (5 item(s))

**Code Locations: Memory leak**

Description	Source	Function	Module	Object Size	Offset	Variable
Allocation site	task_for_lecture5.cpp:141	main	lab3.exe	8		block allocated at task_for_lecture5.cpp:141
139	const size_t numb_cols = 3;					
140						
141	double** matrix = new double*[numb_rows];					
142	for ( size_t i = 0; i < numb_rows; ++i )					
143	{					

**Timeline**

RtlQueryEnvironmentVariable (121840)

Везде видно, что обнаружена утечка памяти.

Проверим теперь наличие или отсутствие блокировок и гонки памяти.

**Intel Inspector** Detect Deadlocks

Target Analysis Type Collection Log Summary

**Problems**

No Problems Detected

Intel Inspector detected no problems at this analysis scope. If this result is unexpected, try rerunning the target using an analysis type with a wider scope. Press F1 for more information.

**Filters**

Severity

Type

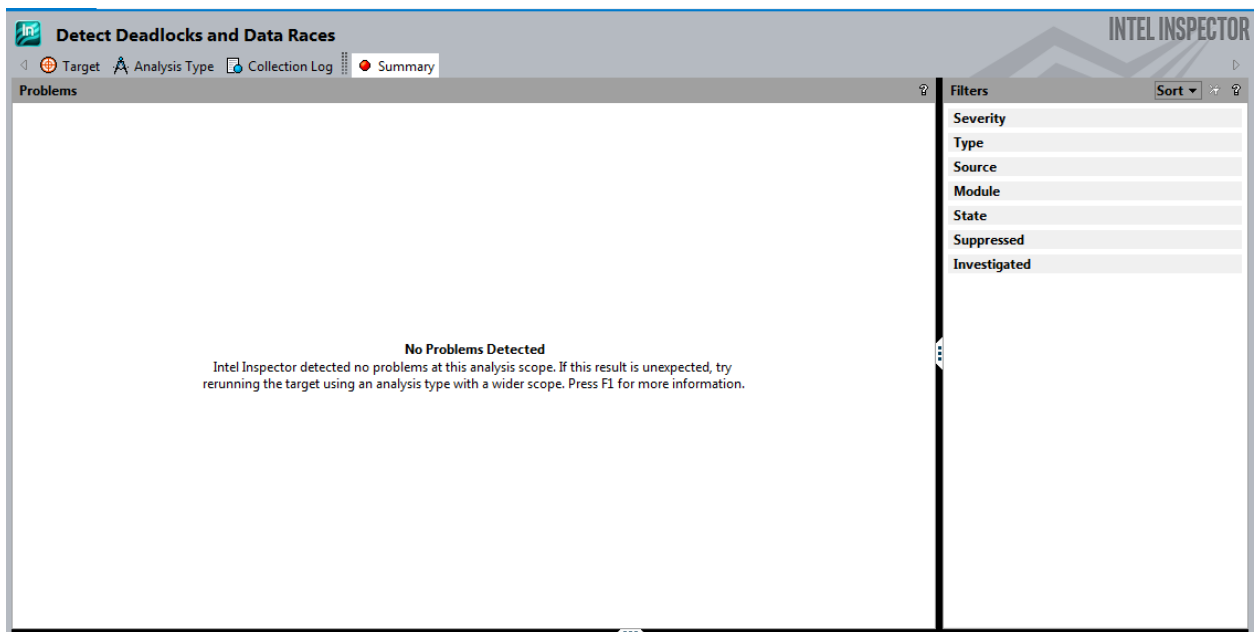
Source

Module

State

Suppressed

Investigated



Проблем не обнаружено.

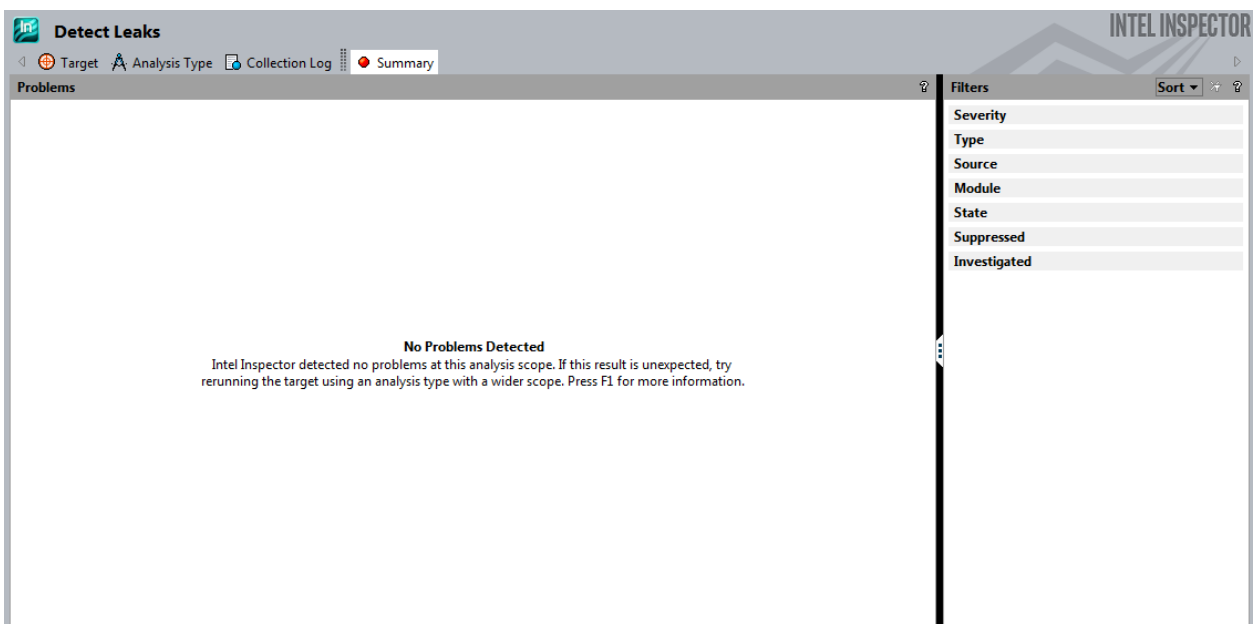
В следующем пункте необходимо будет исправить ошибки, связанные только с утечкой памяти.

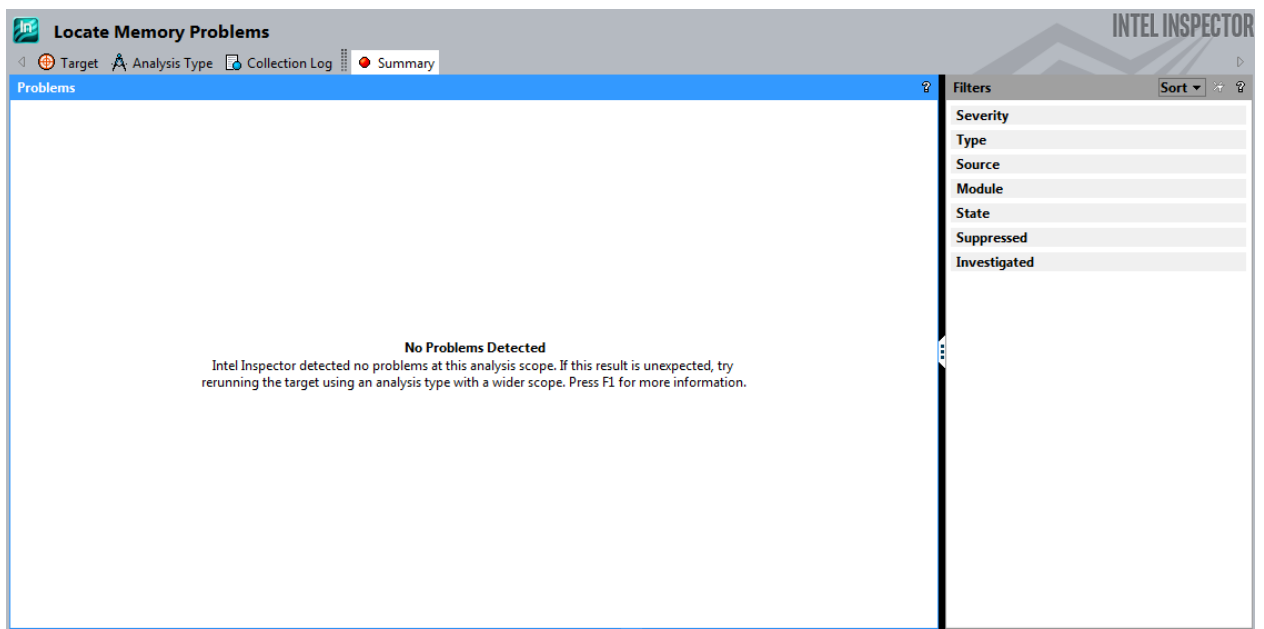
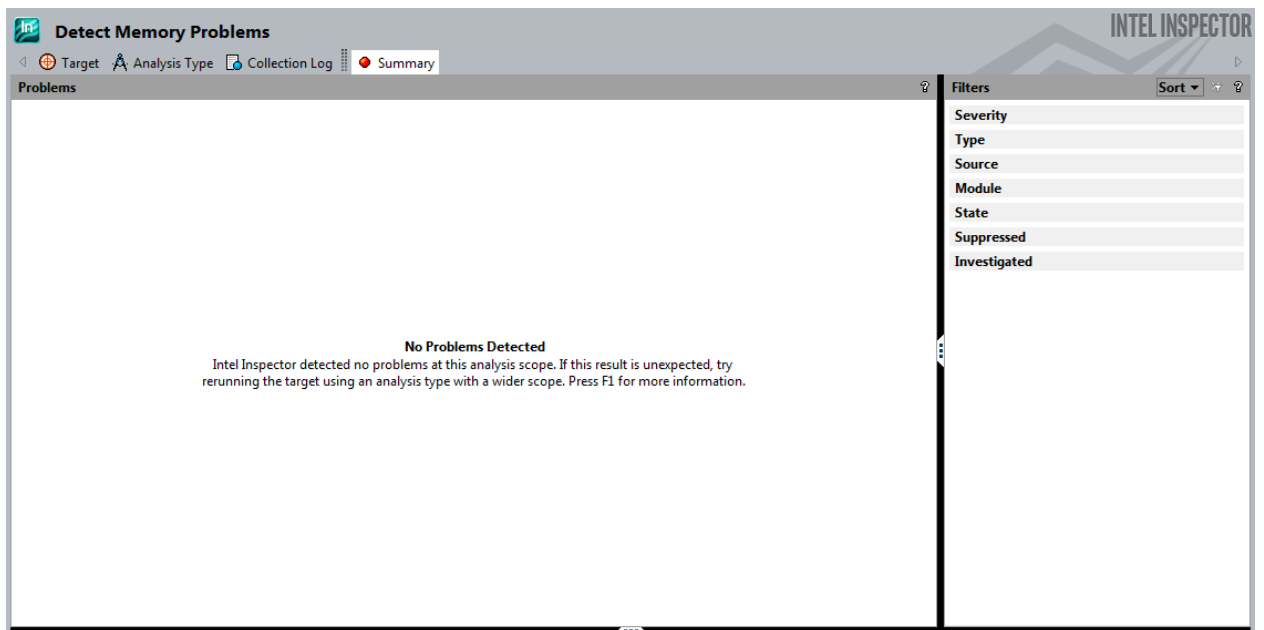
- Измените код программы таким образом, чтобы **Inspector** при проверке не находил в программе ошибок, перечисленных в п. 3. Сделайте скрины результатов запуска **Parallel Inspector**.

```
delete[] average_vals_in_cols;
delete[] average_vals_in_rows;

for (size_t i = 0; i < numb_rows; ++i)
    delete[] matrix[i];
delete[] matrix;
```

Проведем проверку также в трех режимах:





*Все ошибки исправлены.*