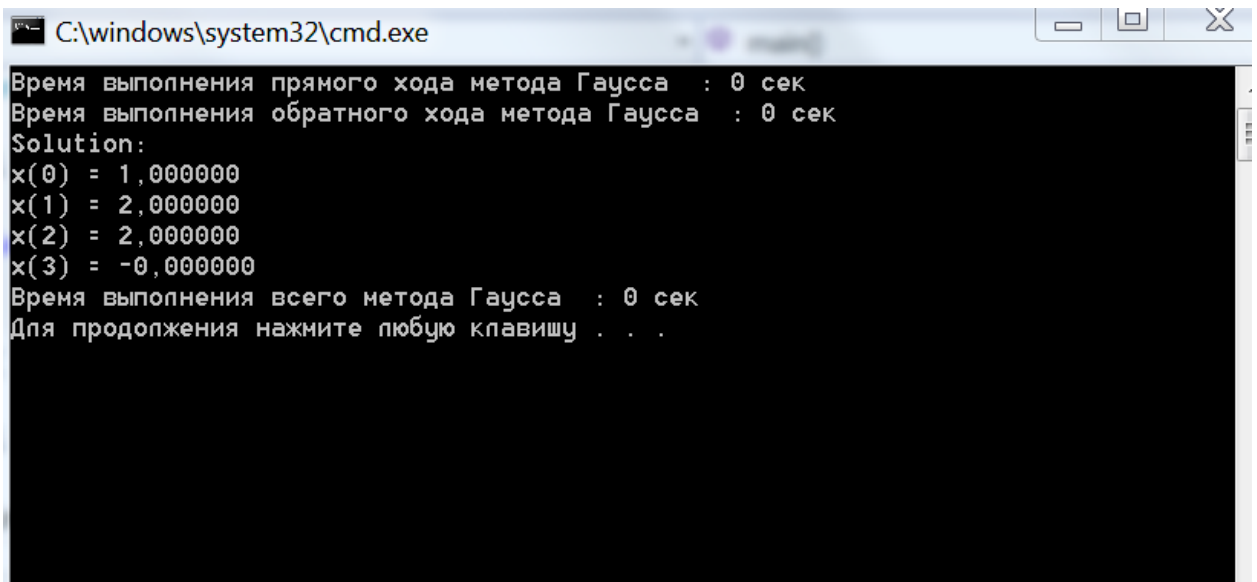


2. Запустите первоначальную версию программы и получите решение для тестовой матрицы `test_matrix`, убедитесь в правильности приведенного алгоритма. Добавьте строки кода для измерения времени (см. задание к занятию 2) выполнения прямого хода метода Гаусса в функцию `SerialGaussMethod()`. Заполните матрицу с количеством строк `MATRIX_SIZE` случайными значениями, используя функцию `InitMatrix()`. Найдите решение СЛАУ для этой матрицы. (Закомментируйте строки кода, где используется тестовая матрица `test_matrix`).



```
C:\windows\system32\cmd.exe
Время выполнения прямого хода метода Гаусса : 0 сек
Время выполнения обратного хода метода Гаусса : 0 сек
Solution:
x(0) = 1,000000
x(1) = 2,000000
x(2) = 2,000000
x(3) = -0,000000
Время выполнения всего метода Гаусса : 0 сек
Для продолжения нажмите любую клавишу . . .
```

```
>> A
```

```
A =
```

```
    2    5    4    1
    1    3    2    1
    2   10    9    7
    3    8    9    2
```

```
>> B
```

```
B =
```

```
    20
    11
    40
    37
```

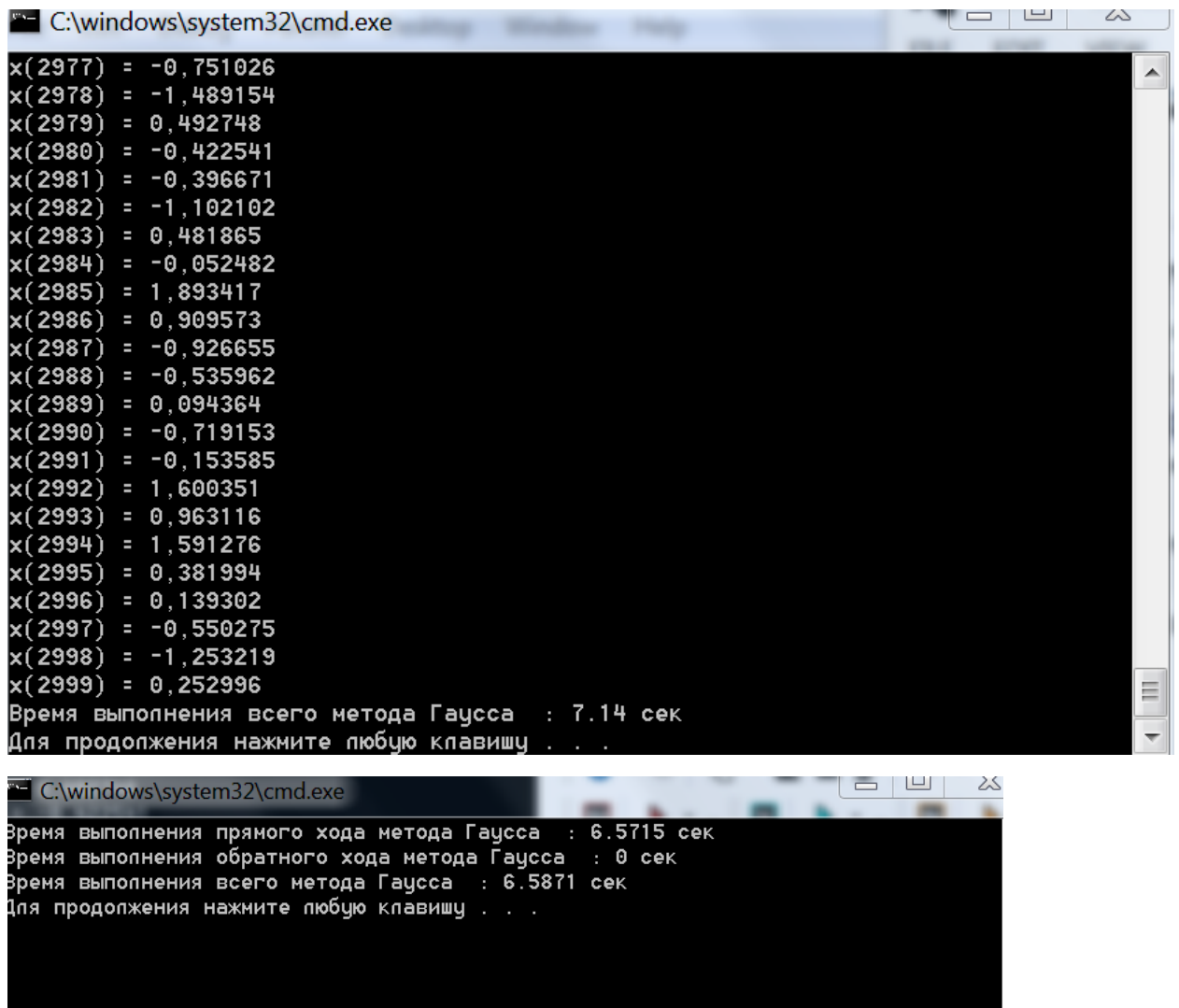
```
>> X=A^(-1)*B
```

```
X =
```

```
    1.0000
    2.0000
    2.0000
   -0.0000
```

Решение тестовой матрицы верно

Решение произвольной матрицы (последовательный метод Гаусса)

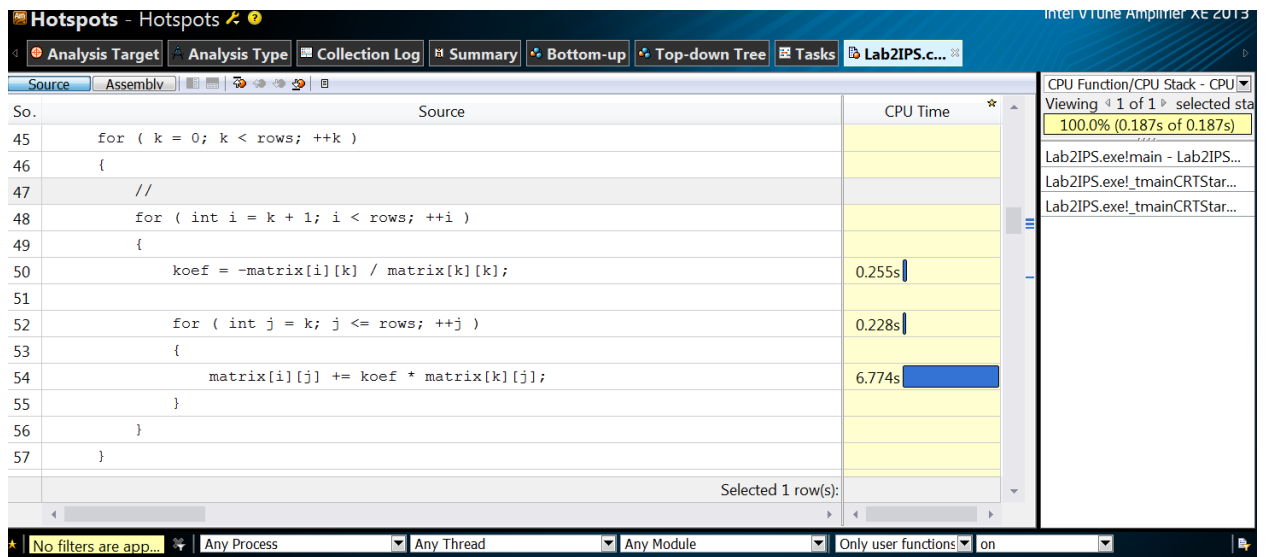


```
C:\windows\system32\cmd.exe
x(2977) = -0,751026
x(2978) = -1,489154
x(2979) = 0,492748
x(2980) = -0,422541
x(2981) = -0,396671
x(2982) = -1,102102
x(2983) = 0,481865
x(2984) = -0,052482
x(2985) = 1,893417
x(2986) = 0,909573
x(2987) = -0,926655
x(2988) = -0,535962
x(2989) = 0,094364
x(2990) = -0,719153
x(2991) = -0,153585
x(2992) = 1,600351
x(2993) = 0,963116
x(2994) = 1,591276
x(2995) = 0,381994
x(2996) = 0,139302
x(2997) = -0,550275
x(2998) = -1,253219
x(2999) = 0,252996
Время выполнения всего метода Гаусса : 7.14 сек
Для продолжения нажмите любую клавишу . . .

C:\windows\system32\cmd.exe
Время выполнения прямого хода метода Гаусса : 6.5715 сек
Время выполнения обратного хода метода Гаусса : 0 сек
Время выполнения всего метода Гаусса : 6.5871 сек
Для продолжения нажмите любую клавишу . . .
```

2. С помощью инструмента Amplifier XE определите наиболее часто используемые участки кода новой версии программы. Сохраните скриншот результатов анализа Amplifier XE. Создайте, на основе последовательной функции SerialGaussMethod(), новую функцию, реализующую параллельный метод Гаусса. Введите параллелизм в новую функцию, используя `cilk_for`. **Примечание:** произвести параллелизацию одного внутреннего цикла прямого хода метода Гаусса (определить какого именно), и внутреннего цикла обратного хода. Время выполнения по-прежнему измерять только для прямого хода.

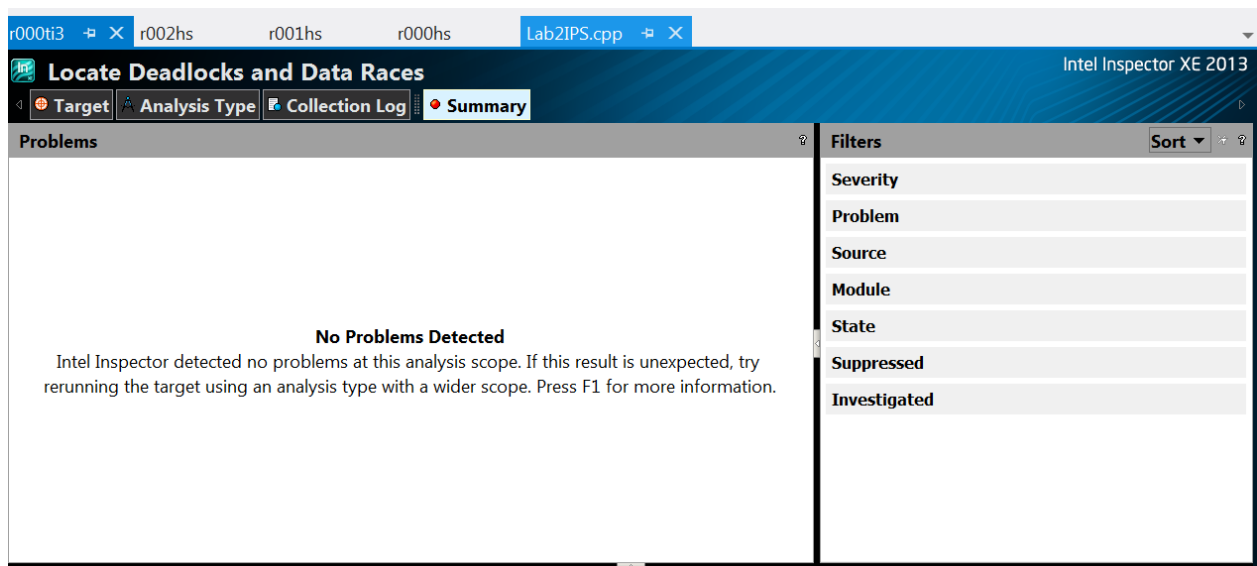
Amplifier XE (последовательный метод Гаусса)



147	// массив решений СЛАУ	
148	double *result = new double[test_matrix_lines];	
149	// инициализация тестовой матрицы	
150	InitMatrix(test_matrix);	109.230ms
151	//test_matrix[0][0] = 2; test_matrix[0][1] = 5; test_matrix[0][2] = 4; test.	
152	//test_matrix[1][0] = 1; test_matrix[1][1] = 3; test_matrix[1][2] = 2; test.	
153	//test_matrix[2][0] = 2; test_matrix[2][1] = 10; test_matrix[2][2] = 9; test.	
154	//test_matrix[3][0] = 3; test_matrix[3][1] = 8; test_matrix[3][2] = 9; test.	
155		
156	high_resolution_clock::time_point t1 = high_resolution_clock::now();	

167	printf("Solution:\n");	
168		
169	for (i = 0; i < test_matrix_lines; ++i)	
170	{	
171	printf("x(%d) = %lf\n", i, result[i]);	78.001ms
172	}	
173	cout << "Время выполнения всего метода Гаусса : " << duration.count() << " с.	
174	delete[] result;	
175		
176	return 0;	
177	}	

- Далее, используя Inspector XE, определите те данные (если таковые имеются), которые принимают участие в гонке данных или в других основных ошибках, возникающих при разработке параллельных программ, и устраните эти ошибки. Сохраните скриншоты анализов, проведенных инструментом Inspector XE: в случае обнаружения ошибок и после их устранения.



2. Убедитесь на примере тестовой матрицы `test_matrix` в том, что функция, реализующая параллельный метод Гаусса работает правильно. Сравните время выполнения прямого хода метода Гаусса для последовательной и параллельной реализации при решении матрицы, имеющей количество строк `MATRIX_SIZE`, заполняющейся случайными числами. Запускайте проект в режиме Release, предварительно убедившись, что включена оптимизация (Optimization->Optimization=/O2). Подсчитайте ускорение параллельной версии в сравнении с последовательной. Выводите значения ускорения на консоль.

Запустим тестовую матрицу с параллельным методом Гаусса

 The screenshot shows a Windows command prompt window titled 'C:\windows\system32\cmd.exe'. The output of the program is displayed in Russian. It shows the execution time for the direct and reverse steps of the Gaussian method, both as 0 seconds. It then displays the solution for a system of linear equations: $x(0) = 1,000000$, $x(1) = 2,000000$, $x(2) = 2,000000$, and $x(3) = -0,000000$. Finally, it shows the total execution time for the Gaussian method as 0 seconds and prompts the user to press any key to continue.


```

C:\windows\system32\cmd.exe
Время выполнения прямого хода метода Гаусса : 0 сек
Время выполнения обратного хода метода Гаусса : 0 сек
Solution:
x(0) = 1,000000
x(1) = 2,000000
x(2) = 2,000000
x(3) = -0,000000
Время выполнения всего метода Гаусса : 0 сек
Для продолжения нажмите любую клавишу . . .
  
```

Решение совпадает

Сравним время выполнения параллельной и послед реализации при матрице размера 5000 на 5000

```
C:\windows\system32\cmd.exe
Последовательная реализация :
Время выполнения прямого хода метода Гаусса : 73.0441 сек
Время выполнения обратного хода метода Гаусса : 0.02 сек
Параллельная реализация :
Время выполнения прямого хода метода Гаусса : 32.3068 сек
Время выполнения обратного хода метода Гаусса : 0.0725 сек
Для продолжения нажмите любую клавишу . . .
```

Ускорение программы :

```
C:\windows\system32\cmd.exe
Последовательная реализация :
Время выполнения прямого хода метода Гаусса : 70.485 сек
Время выполнения обратного хода метода Гаусса : 0.019 сек
Параллельная реализация :
Время выполнения прямого хода метода Гаусса : 30.565 сек
Время выполнения обратного хода метода Гаусса : 0.064 сек
Ускорение работы программы : 2.30607
Для продолжения нажмите любую клавишу . . .
```