

Задание: Инициализировать массивы  $A$  и  $B$  в соответствии с выражением:

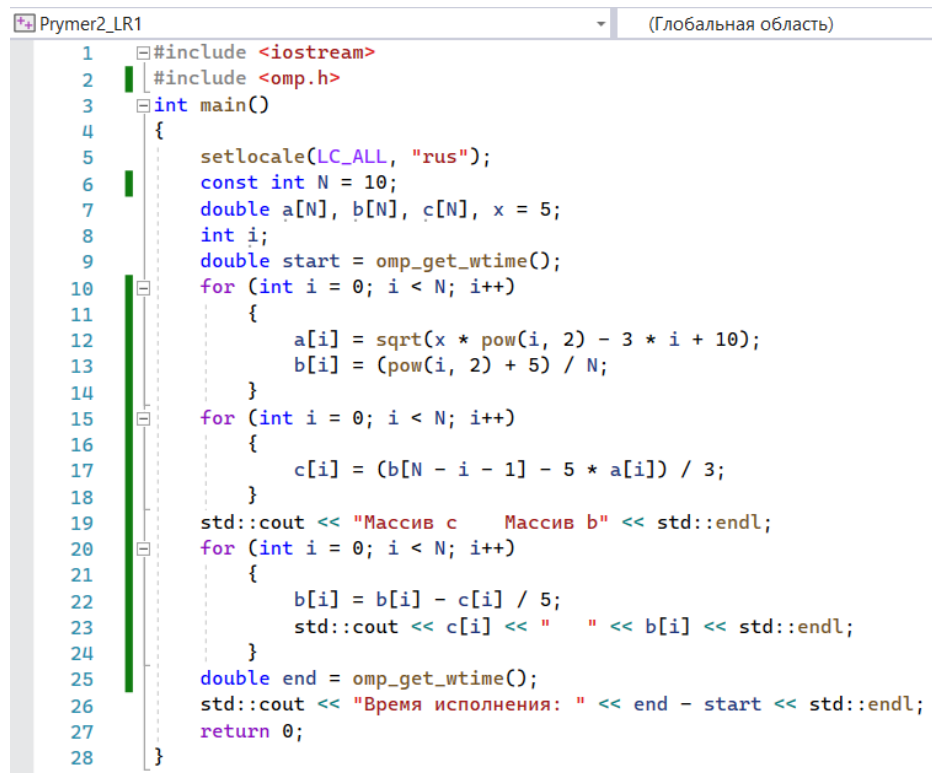
$$A[i] = \sqrt{x \cdot i^2 - 3i + 10}, \quad B[i] = \frac{i^2 + 5}{N}, \quad \text{где } x - \text{произвольная константа (определяется пользователем),}$$

$N$  – количество элементов массива.

Выполнить вычисление массива  $C$  и переопределение массива  $B$ :

$$C[i] = \frac{b[N-i-1] - 5 \cdot a[i]}{3}, \quad B[i] = B[i] - \frac{C[i]}{5}.$$

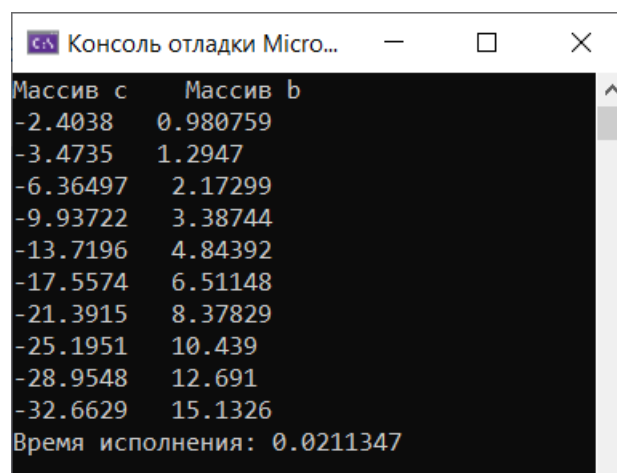
Напишем последовательную программу на языке C++. В программе предусмотрим замер времени выполнения вычислений (без учета времени ввода-вывода) (рис. 1).



```
1 #include <iostream>
2 #include <omp.h>
3 int main()
4 {
5     setlocale(LC_ALL, "rus");
6     const int N = 10;
7     double a[N], b[N], c[N], x = 5;
8     int i;
9     double start = omp_get_wtime();
10    for (int i = 0; i < N; i++)
11    {
12        a[i] = sqrt(x * pow(i, 2) - 3 * i + 10);
13        b[i] = (pow(i, 2) + 5) / N;
14    }
15    for (int i = 0; i < N; i++)
16    {
17        c[i] = (b[N - i - 1] - 5 * a[i]) / 3;
18    }
19    std::cout << "Массив c    Массив b" << std::endl;
20    for (int i = 0; i < N; i++)
21    {
22        b[i] = b[i] - c[i] / 5;
23        std::cout << c[i] << "    " << b[i] << std::endl;
24    }
25    double end = omp_get_wtime();
26    std::cout << "Время исполнения: " << end - start << std::endl;
27    return 0;
28 }
```

Рисунок 1 – Последовательная программа

Результат исполнения кода представлен на рисунке 2.



```
Консоль отладки Micro...
Массив c    Массив b
-2.4038    0.980759
-3.4735    1.2947
-6.36497    2.17299
-9.93722    3.38744
-13.7196    4.84392
-17.5574    6.51148
-21.3915    8.37829
-25.1951    10.439
-28.9548    12.691
-32.6629    15.1326
Время исполнения: 0.0211347
```

Рисунок 2 – Результат исполнения последовательного кода

Проанализируем время выполнения программы и загрузку процессора при различных значениях *N* (табл. 1). При этом удалим из программного кода этап вывода массивов на экран.

Таблица 1 – Анализ исполнения последовательного кода при различных размерах массивов

N	Время исполнения, с	Загрузка процессора																																				
100	2.73e-05	<div><div>Сеанс диагностики: 3,037 с</div><div><div></div><div>ЦП (% всех процессоров)</div><div></div></div><div>Ключевая аналитика</div><div>Аналитические сведения не найдены.</div><div>Важнейшие функции</div><table><thead><tr><th>Имя функции</th><th>Общее время ЦП (единицы, %)</th><th>Собственное время ЦП (единицы, %)</th></tr></thead><tbody><tr><td>[Внешний вызов] ucrtbased.dll!0x00007fe162a1793</td><td>24 (66,67 %)</td><td>24 (66,67 %)</td></tr><tr><td>[Внешний вызов] ntdll.dll!0x00007feebcc4c1e</td><td>7 (19,44 %)</td><td>7 (19,44 %)</td></tr><tr><td>[Внешний вызов] ucrtbased.dll!0x00007fe162b41f6</td><td>2 (5,56 %)</td><td>2 (5,56 %)</td></tr><tr><td>[Внешний вызов] ntdll.dll!0x00007feebcc2680</td><td>1 (2,78 %)</td><td>1 (2,78 %)</td></tr><tr><td>0x000000003f6684fa</td><td>1 (2,78 %)</td><td>1 (2,78 %)</td></tr></tbody></table><div>Критический путь</div><table><thead><tr><th>Имя функции</th><th>Общее время ЦП (единицы, %)</th><th>Собственное время ЦП (единицы, %)</th></tr></thead><tbody><tr><td>Pymer4_LR1 (PID: 18036)</td><td>36 (100,00 %)</td><td>0 (0,00 %)</td></tr><tr><td>[Системный код] ntdll.dll!0x00007feebcc26a1</td><td>26 (72,22 %)</td><td>0 (0,00 %)</td></tr><tr><td>mainCRTStartup</td><td>26 (72,22 %)</td><td>0 (0,00 %)</td></tr><tr><td>static int __sort_common_main()</td><td>26 (72,22 %)</td><td>0 (0,00 %)</td></tr><tr><td>static int __sort_common_main_seh()</td><td>26 (72,22 %)</td><td>0 (0,00 %)</td></tr></tbody></table><div>Пять ведущих категорий</div><div><div>Ядро : 94,4 % (34)</div><div>Другое : 5,6 % (2)</div></div></div>	Имя функции	Общее время ЦП (единицы, %)	Собственное время ЦП (единицы, %)	[Внешний вызов] ucrtbased.dll!0x00007fe162a1793	24 (66,67 %)	24 (66,67 %)	[Внешний вызов] ntdll.dll!0x00007feebcc4c1e	7 (19,44 %)	7 (19,44 %)	[Внешний вызов] ucrtbased.dll!0x00007fe162b41f6	2 (5,56 %)	2 (5,56 %)	[Внешний вызов] ntdll.dll!0x00007feebcc2680	1 (2,78 %)	1 (2,78 %)	0x000000003f6684fa	1 (2,78 %)	1 (2,78 %)	Имя функции	Общее время ЦП (единицы, %)	Собственное время ЦП (единицы, %)	Pymer4_LR1 (PID: 18036)	36 (100,00 %)	0 (0,00 %)	[Системный код] ntdll.dll!0x00007feebcc26a1	26 (72,22 %)	0 (0,00 %)	mainCRTStartup	26 (72,22 %)	0 (0,00 %)	static int __sort_common_main()	26 (72,22 %)	0 (0,00 %)	static int __sort_common_main_seh()	26 (72,22 %)	0 (0,00 %)
Имя функции	Общее время ЦП (единицы, %)	Собственное время ЦП (единицы, %)																																				
[Внешний вызов] ucrtbased.dll!0x00007fe162a1793	24 (66,67 %)	24 (66,67 %)																																				
[Внешний вызов] ntdll.dll!0x00007feebcc4c1e	7 (19,44 %)	7 (19,44 %)																																				
[Внешний вызов] ucrtbased.dll!0x00007fe162b41f6	2 (5,56 %)	2 (5,56 %)																																				
[Внешний вызов] ntdll.dll!0x00007feebcc2680	1 (2,78 %)	1 (2,78 %)																																				
0x000000003f6684fa	1 (2,78 %)	1 (2,78 %)																																				
Имя функции	Общее время ЦП (единицы, %)	Собственное время ЦП (единицы, %)																																				
Pymer4_LR1 (PID: 18036)	36 (100,00 %)	0 (0,00 %)																																				
[Системный код] ntdll.dll!0x00007feebcc26a1	26 (72,22 %)	0 (0,00 %)																																				
mainCRTStartup	26 (72,22 %)	0 (0,00 %)																																				
static int __sort_common_main()	26 (72,22 %)	0 (0,00 %)																																				
static int __sort_common_main_seh()	26 (72,22 %)	0 (0,00 %)																																				

```
1 #include <iostream>
2 #include <omp.h>
3 int main()
4 {
5     setlocale(LC_ALL, "rus");
6     const int N = 100;
7     double a[N], b[N], c[N], x = 5;
8     int i;
9     double start = omp_get_wtime();
10    #pragma omp parallel num_threads (N) private(i)
11    {
12        i = omp_get_thread_num();
13        a[i] = sqrt(x * pow(i, 2) - 3 * i + 10);
14        b[i] = (pow(i, 2) + 5) / N;
15    }
16    #pragma omp parallel num_threads (N) private(i)
17    {
18        i = omp_get_thread_num();
19        c[i] = (b[N - i - 1] - 5 * a[i]) / 3;
20    }
21    #pragma omp parallel num_threads (N) private(i)
22    {
23        i = omp_get_thread_num();
24        b[i] = b[i] - c[i] / 5;
25    }
26    double end = omp_get_wtime();
27    std::cout << "Время исполнения: " << end - start << std::endl;
28    return 0;
29 }
```

Рисунок 2 – Параллельный код

Проанализируем время выполнения программы и загрузку процессора при различных значениях *N* (табл. 2).

Таблица 2 – Анализ исполнения параллельного кода при различных размерах массивов

N	Время исполнения, с	Загрузка процессора																														
100	0.0053167	<div><div>Сеанс диагностики: 2,71 с</div><div><div>ЦП (% всех процессоров)</div><div></div></div><div>Ключевая аналитика</div><div>Аналитические сведения не найдены.</div><div>Важнейшие функции</div><table><thead><tr><th>Имя функции</th><th>Общее время ЦП [единицы, %]</th><th>Собственное время ЦП [единицы, %]</th></tr></thead><tbody><tr><td>[Внешний вызов] ntdll.dll!0x00007f6ebcc26a1</td><td>72 (69,90 %)</td><td>44 (42,72 %)</td></tr><tr><td>[Внешний вызов] ucrtbase.dll!0x00007f6e977c2e7</td><td>24 (23,30 %)</td><td>24 (23,30 %)</td></tr><tr><td>0xfffff8072500aab8</td><td>7 (6,80 %)</td><td>7 (6,80 %)</td></tr><tr><td>[Внешний вызов] ntdll.dll!0x00007f6ebce4c1e</td><td>6 (5,83 %)</td><td>6 (5,83 %)</td></tr><tr><td>0xfffff8072500ab5f</td><td>4 (3,88 %)</td><td>4 (3,88 %)</td></tr></tbody></table><div>Критический путь</div><table><thead><tr><th>Имя функции</th><th>Общее время ЦП [единицы, %]</th><th>Собственное время ЦП [единицы, %]</th></tr></thead><tbody><tr><td>Pymer3_LR1 (PID: 1800)</td><td>103 (100,00 %)</td><td>10 (9,71 %)</td></tr><tr><td>[Системный код] ntdll.dll!0x00007f6ebcc26a1</td><td>72 (69,90 %)</td><td>44 (42,72 %)</td></tr><tr><td>static int _sclr_common_main_seh()</td><td>27 (26,21 %)</td><td>0 (0,00 %)</td></tr></tbody></table><div>Пять ведущих категорий</div><div><div>Ядро : 50,5 % (52)</div><div>Среда выполнения : 26,2 % (27)</div><div>Другое : 23,3 % (24)</div></div></div>	Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]	[Внешний вызов] ntdll.dll!0x00007f6ebcc26a1	72 (69,90 %)	44 (42,72 %)	[Внешний вызов] ucrtbase.dll!0x00007f6e977c2e7	24 (23,30 %)	24 (23,30 %)	0xfffff8072500aab8	7 (6,80 %)	7 (6,80 %)	[Внешний вызов] ntdll.dll!0x00007f6ebce4c1e	6 (5,83 %)	6 (5,83 %)	0xfffff8072500ab5f	4 (3,88 %)	4 (3,88 %)	Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]	Pymer3_LR1 (PID: 1800)	103 (100,00 %)	10 (9,71 %)	[Системный код] ntdll.dll!0x00007f6ebcc26a1	72 (69,90 %)	44 (42,72 %)	static int _sclr_common_main_seh()	27 (26,21 %)	0 (0,00 %)
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]																														
[Внешний вызов] ntdll.dll!0x00007f6ebcc26a1	72 (69,90 %)	44 (42,72 %)																														
[Внешний вызов] ucrtbase.dll!0x00007f6e977c2e7	24 (23,30 %)	24 (23,30 %)																														
0xfffff8072500aab8	7 (6,80 %)	7 (6,80 %)																														
[Внешний вызов] ntdll.dll!0x00007f6ebce4c1e	6 (5,83 %)	6 (5,83 %)																														
0xfffff8072500ab5f	4 (3,88 %)	4 (3,88 %)																														
Имя функции	Общее время ЦП [единицы, %]	Собственное время ЦП [единицы, %]																														
Pymer3_LR1 (PID: 1800)	103 (100,00 %)	10 (9,71 %)																														
[Системный код] ntdll.dll!0x00007f6ebcc26a1	72 (69,90 %)	44 (42,72 %)																														
static int _sclr_common_main_seh()	27 (26,21 %)	0 (0,00 %)																														

10000

0.699417

