

НИУ ВШЭ ФКН  
Архитектура Вычислительных Систем  
Домашнее задание №4  
Иванова Виолетта Александровна  
Студентка БПИ-195(1)  
ВАРИАНТ №11

**Условие задачи:**

Определить индексы  $i, j$  ( $i \neq j$ ), для которых выражение  $A[i] - A[i+1] + A[i+2] - A[i+3] \pm \dots A[j]$  имеет максимальное значение. Входные данные: массив чисел  $A$ , произвольной длины большей 10. Количество потоков не является параметром задачи.

**Ход решения:**

Выписав формулу и рассмотрев некоторые варианты – я приняла решение использовать модель *итеративного параллелизма* в рамках решения этой задачи.

Программа позволяет просчитать значения всех выражений с соответствующими индексами.

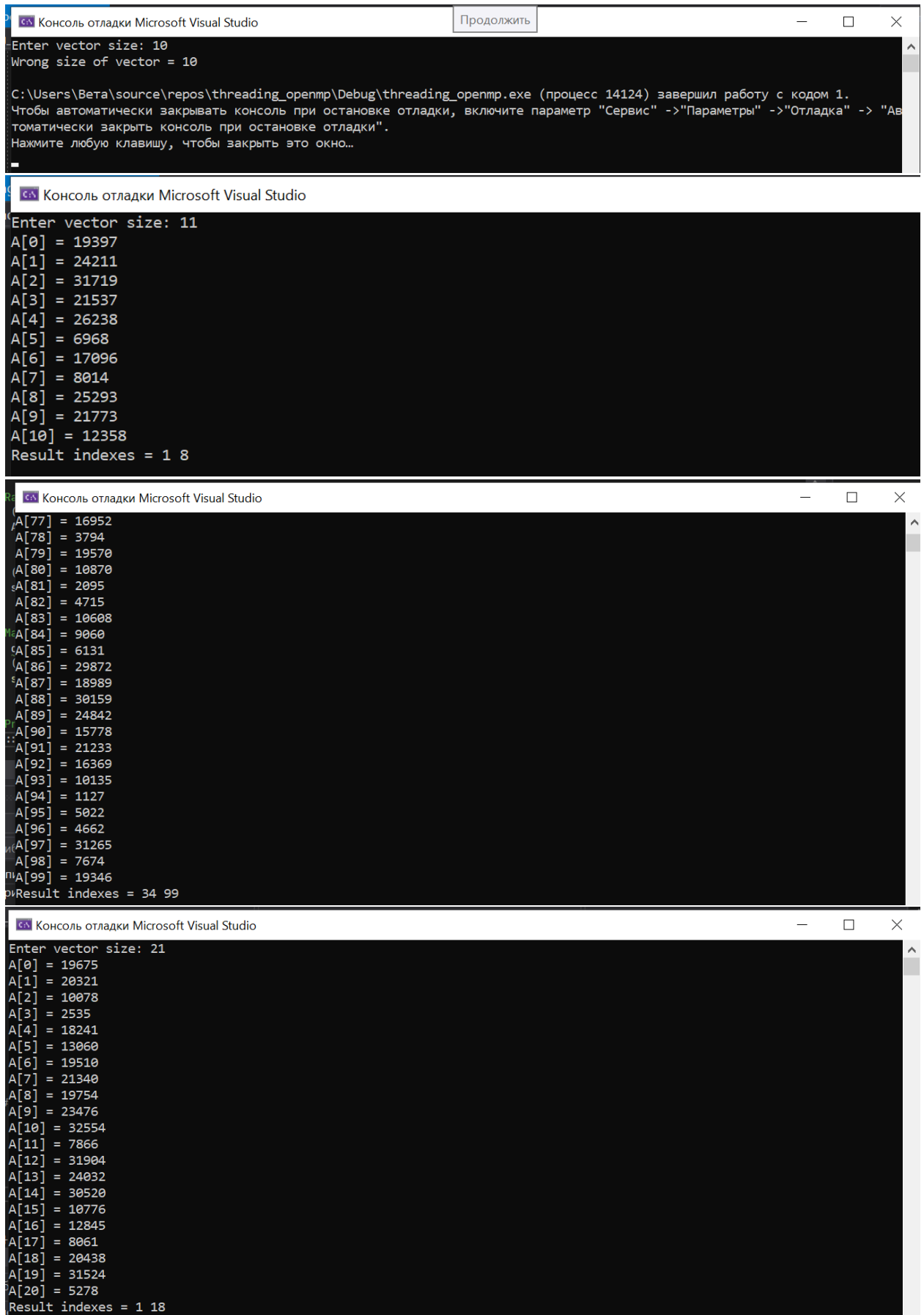
Результаты вычислений по итогам вычисления значения для *каждого индекса* сохраняются в векторе векторов чисел  $C$ . Так, например, выражение от  $A[i] \dots A[j]$  хранится в ячейке  $C[i][j]$ . После этого происходит сравнение значения с глобальной переменной и присваивание индексов, если значение выражения оказалось больше, чем было раньше.

Благодаря такому алгоритму, потоки никак не влияют друг на друга, но орентр позволяет потокам работать с тремя глобальными переменными – максимальное значение и два индекса.

Работа потоков происходит под руководством библиотеки, она сама определяет подходящее число потоков.

Изначальный массив заполняется случайными числами, возможности переполнения исключены.

## Примеры работы программы на разных входных данных



```
Консоль отладки Microsoft Visual Studio
Enter vector size: 10
Wrong size of vector = 10

C:\Users\Bera\source\repos\threading_openmp\Debug\threading_openmp.exe (процесс 14124) завершил работу с кодом 1.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрывать консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

```
Консоль отладки Microsoft Visual Studio
Enter vector size: 11
A[0] = 19397
A[1] = 24211
A[2] = 31719
A[3] = 21537
A[4] = 26238
A[5] = 6968
A[6] = 17096
A[7] = 8014
A[8] = 25293
A[9] = 21773
A[10] = 12358
Result indexes = 1 8
```

```
Консоль отладки Microsoft Visual Studio
A[77] = 16952
A[78] = 3794
A[79] = 19570
A[80] = 10870
A[81] = 2095
A[82] = 4715
A[83] = 10608
A[84] = 9060
A[85] = 6131
A[86] = 29872
A[87] = 18989
A[88] = 30159
A[89] = 24842
A[90] = 15778
A[91] = 21233
A[92] = 16369
A[93] = 10135
A[94] = 1127
A[95] = 5022
A[96] = 4662
A[97] = 31265
A[98] = 7674
A[99] = 19346
Result indexes = 34 99
```

```
Консоль отладки Microsoft Visual Studio
Enter vector size: 21
A[0] = 19675
A[1] = 20321
A[2] = 10078
A[3] = 2535
A[4] = 18241
A[5] = 13060
A[6] = 19510
A[7] = 21340
A[8] = 19754
A[9] = 23476
A[10] = 32554
A[11] = 7866
A[12] = 31904
A[13] = 24032
A[14] = 30520
A[15] = 10776
A[16] = 12845
A[17] = 8061
A[18] = 20438
A[19] = 31524
A[20] = 5278
Result indexes = 1 18
```

## Текст программы:

```
#include <iostream>
#include <omp.h>
#include <vector>
#include <thread>

unsigned int arrSize;
std::vector<std::vector<double>>> C;
std::vector<double> A;
double result = DBL_MIN;
int result_i = -1, result_j = -1;

void summary(int index_i) {
    // int index_i = (int)param;
    for (unsigned int j = index_i; j < arrSize; j++) {
        if (j == index_i) {
            C[index_i][j] = A[j];
        }

        else {
            C[index_i][j] = C[index_i][j - 1] - A[j] * pow(-1, j + index_i);
        }

        if (result < C[index_i][j]) {
            result = C[index_i][j];
            result_i = index_i;
            result_j = j;
        }
    }
}

int main() {
    // Init rand.
    srand(time(NULL));

    // Enter size of array.
    std::cout << "Enter vector size: ";
    std::cin >> arrSize;

    // Check entering.
    if (arrSize <= 10 || arrSize > INT32_MAX) {
        std::cout << "Wrong size of vector = " << arrSize << "\n";
        return 1;
    }

    A.resize(arrSize, 0);
    C.resize(arrSize, std::vector<double>(arrSize, DBL_MIN));

    // Random array values.
    for (unsigned int i = 0; i < arrSize; ++i) {
        A[i] = 1 + rand() % 100000;
    }

    for (unsigned int i = 0; i < arrSize; i++) {
        std::cout << "A[" << i << "] = " << A[i] << "\n";
    }

    // Make parallel computations.
#pragma omp parallel for shared(result, result_i, result_j)
    for (int i = 0; i < arrSize - 1; i++) {
        summary(i);
    }

    // Print result.
    std::cout << "Result indexes = " << result_i << " " << result_j << "\n";

    return 0;
}
```