

Вариант №12

Определить индексы i, j , для которых существует наиболее длинная последовательность $A[i] < A[i+1] < A[i+2] < A[i+3] < \dots < A[j]$. Входные данные: массив чисел A , произвольной длины большей 1000. Количество потоков

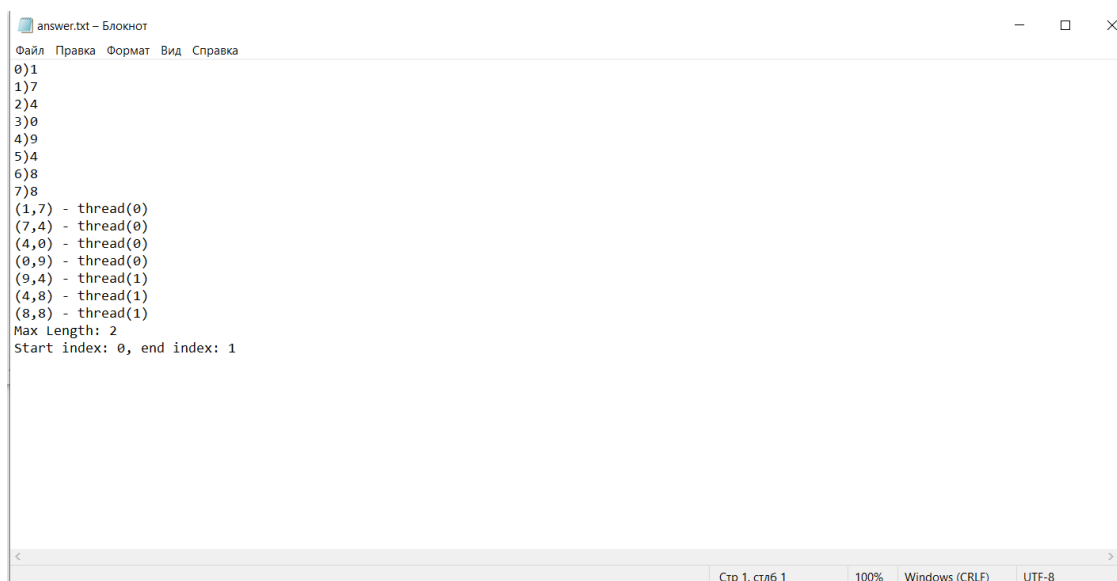
1. Модель

При разработке программы мною была выбрана итеративная модель[1] параллелизма, так как при для решения задачи удобно создать равноправные потоки, которые выполняют практически одинаковые задачи(в данном случае – сравнение двух соседних элементов массива). Данная задача похожа на задачу поиска суммы квадратов элементов массива[2].

Входные данные поступает в программу в качестве аргументов командной строки. Первый параметр – размер массива. Второй параметр – количество потоков. Третий параметр – файл, в который выводится информация о решении задачи. При некорректных входных данных в консоль пишется соответствующее сообщение об ошибке. Элементы массива выводятся в файл по одному в строке с указанием индекса элемента. Для удобства все элементы не превосходят 9. После этого в файл выводится информация о том, какие два элемента сравниваются и в каком потоке это происходит. В конце, когда все потоки выполняются, в файл выводится максимальная длина возрастающей последовательности и индексы начального и конечного элементов этой последовательности.

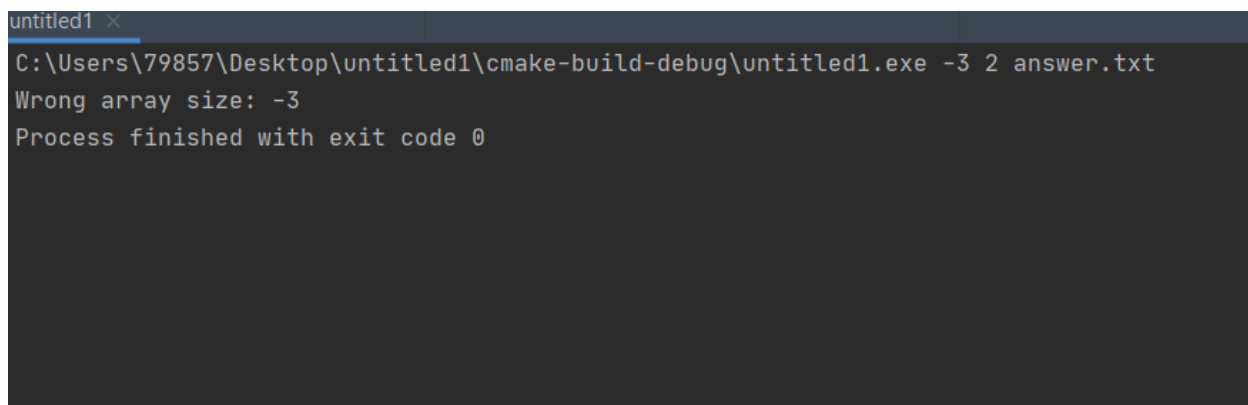
2. Тестирование

Входные данные: 8 2 answer.txt



```
answer.txt – Блокнот
Файл  Правка  Формат  Вид  Справка
0)1
1)7
2)4
3)0
4)9
5)4
6)8
7)8
(1,7) - thread(0)
(7,4) - thread(0)
(4,0) - thread(0)
(0,9) - thread(0)
(9,4) - thread(1)
(4,8) - thread(1)
(8,8) - thread(1)
Max Length: 2
Start index: 0, end index: 1
```

Пример работы программы при некорректных входных данных:



```
untitled1 x
C:\Users\79857\Desktop\untitled1\cmake-build-debug\untitled1.exe -3 2 answer.txt
Wrong array size: -3
Process finished with exit code 0
```

3. Список используемых источников

Архитектура параллельных вычислительных систем. Многопоточность
<http://softcraft.ru/edu/comparch/lect/07-parthread/>

2) Многопоточность. Простая многопоточная программа. Основные функции
<http://softcraft.ru/edu/comparch/practice/thread/01-simple/>

3) Многопоточное программирование.OpenMP
<http://softcraft.ru/edu/comparch/practice/thread/03-openmp/>