Лабораторная работа №2

Необходимо реализовать **два** **консольных** приложения **под OS Windows на языке C++.** Оба задания (А и Б) необходимо сдавать **одновременно**. Так же необходимо написать краткий **отчет** по выполняемой работе и сдать его в виде **PDF** файла. В отчете должен быть описан математический алгоритм, который вы использовали (его суть, условие завершения вычислений, условие отсутствия решения).

**Внимание**: сдать необходимо два файла с исходным кодом (код каждого из заданий компонуется в одни файл) и файл с отчетом в pdf формате.

**Внимание**: Реализация приложений осуществляется на С++03 (проект MSVC2008) и технологии WinAPI.

## Задание А:

Реализовать программу, которая создает вспомогательный поток, выполняющий некоторые вычисления в фоне, и ожидает его завершения. Приоритет вспомогательного потока не должен превосходить normal. Вспомогательный поток внутри себя исполняет функцию, **аргументами** которой являются имена двух текстовых файлов (конфигурационный и выходной). По завершении вычислений необходимо из главного потока вывести в консоль время, затраченное на вычисления.

Вычисление содержит следующие этапы:

1. Чтение необходимых параметров задачи из конфигурационного текстового файла.
2. Произведение необходимых вычислений (решаемая задача определяется вашим вариантом).
3. Сохранение результата в выходной текстовый файл.

## Задание Б:

Реализовать программу, работающую по нижеприведенному сценарию:

1. Из консоли принимаются два аргумента TaskCount и ThreadCount.
2. Основной поток генерирует TaskCount наборов параметров (для задачи из задания А), и сохраняет их в некоторый контейнер (контейнер с параметрами один для всех потоков). При генерации параметров использовать датчик случайных чисел. Алгоритм генерации должен быть такой, что бы решения всех задач не были одинаковыми (например 0 или No solution).
3. Создается ThreadCount одинаковых вспомогательных потоков, которые берут задания из контейнера, и выполняют их асинхронно (т.е. как только поток выполнил свое текущее задание, он берет следующее и выполняет его. И так, пока все задания не будут выполнены. ). Результаты своих вычислений все потоки складывают в один **общий** контейнер.
4. После того, как все задания будет вычислены, основной поток сохраняет все результаты попарно (параметры задачи, ответ) в выходной текстовый файл.
5. После основным потоком на экран выводится статистика: сколько задач всего решено, сколько не решилось из-за ошибки в процессе вычисления, сколько не имеют решений, сколько задач решил каждый из потоков, сколько времени он на это затратил, минимальное и максимальное время решения задачи, время затраченное на запись данных в файл. Статистика должна быть выведена на экран в удобочитаемом виде.

**Важно** (для каждого задания):

1. В вашей реализации для хранения параметров задачи необходимо создать подходящую структуру, предусмотреть для нее операции ввода/вывода из/в поток.
2. Предусмотреть обработку исключений в коде, исполняемом вспомогательным потоком.

## Дополнительное задание С:

Задание является бонусным и может быть получено студентом, имеющим высокий средний балл в рейтинге. Выполняется по желанию студента, в четко оговоренные сроки.

## Варианты заданий:

|  |  |  |  |
| --- | --- | --- | --- |
| *Вариант* | *Описание задачи* | *Входные данные* | *Выходные данные* |
| 1 | Нахождение минимального и максимального значения функции  **y = (x-3) / (3+x2)** на интервале [**a**, **b**] | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка с минимальным и максимальным значением функции |
| 2 | Нахождение корней функции  **y = sin(x) / (2+3x2)** на интервале [**a**, **b**]  методом хорд | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка со значением хотя бы одного корня, либо строка “No solutions” |
| 3 | Нахождение корней функции  **y = cos(x) / 5x2** на интервале [**a**, **b**]  методом касательных | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка со значением хотя бы одного корня, либо строка “No solutions” |
| 4 | Нахождение корней функции  **y = sin(ax) / (b+x2)** на интервале [**-10**, **10**] методом хорд | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка со значением хотя бы одного корня, либо строка “No solutions” |
| 5 | Нахождение корней функции  **y = a \* cos(x) / bx2** на интервале [**0**, **100**] методом касательных | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка со значением хотя бы одного корня, либо строка “No solutions” |
| 6 | Решение определенного интеграла от функции **sin(ax) / (b+x2)** на интервале [**0**, **100**] методом Симпсона | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка с найденным значением интеграла. |
| 7 | Решение определенного интеграла от функции **a \* cos(x) / bx2** на интервале [1, 2] методом трапеций | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка с найденным значением интеграла. |
| 8 | Нахождение дисперсии и среднего квадратичного отклонения | Первая строка – число наблюдений (от 2 до 1000), во второй через пробел – значения наблюдений (помещаются в тип double) | Строка с найденными значениями дисперсии и среднего квадратичного отклонения |
| 9 | Нахождение для последовательности из N чисел среднего арифметического и среднего геометрического | Первая строка – величина N (от 1 до 1000), во второй через пробел – значения (помещаются в тип double) | Строка с найденными значениями среднего арифметического и среднего геометрического |
| 10 | Решение определенного интеграла от функции **(x-3) / (3+x2)** на интервале [**a**, **b**] методом Симпсона | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка с найденным значением интеграла. |
| 11 | Решение определенного интеграла от функции **cos(x) / 5x2** на интервале [**a**, **b**] методом трапеций | Одна строка со значениями **a, b** и точностью вычислений. Числа разделяются пробелами | Строка с найденным значением интеграла. |
| 12 | Проверить, является ли заданный четырёхугольник выпуклым. | Четыре строки, каждая строка содержит координаты очередной вершины четырёхугольника (два числа через пробел). Вершины перечислены в порядке обхода границы. | 0 – нет, не является;  1 – да, является. |
| 13 | Проверить, принадлежит ли точка заданному четырёхугольнику. | Пять строк, первые четыре строки содержит координаты вершин четырёхугольника (два числа через пробел). Вершины перечислены в порядке обхода границы. Пятая строка – координаты проверяемой точки (также два числа через пробел). | 0 – нет, не принадлежит;  1 – да, принадлежит. |
| 14 | Проверить, является ли заданный пятиугольник выпуклым. | Пять строк, каждая строка содержит координаты очередной вершины пятиугольника (два числа через пробел). Вершины перечислены в порядке обхода границы. | 0 – нет, не является;  1 – да, является. |
| 15 | Проверить, принадлежит ли точка заданному пятиугольнику. | Шесть строк, первые пять строк содержит координаты вершин пятиугольника (два числа через пробел). Вершины перечислены в порядке обхода границы. Шестая строка – координаты проверяемой точки (также два числа через пробел). | 0 – нет, не принадлежит;  1 – да, принадлежит. |
| 16 | Заданы координаты N точек. Образуют ли первые три точки треугольник? Если да, то необходимо определить лежат ли все остальные точки внутри этого треугольника. | Первая строка содержит число N (3 < N < 1000). Следующие строки содержат координаты точек (два числа через пробел). | -1 – первые три точки не образуют треугольник  0 – нет  1 – да, все точки лежат внутри треугольника |
| 17 | Заданы две параллельные прямые и множество точек; определить, лежат ли все точки этого множества между этими двумя прямыми. | Первая строка содержит координаты двух точек, принадлежащих первой прямой (четыре числа через пробел: x1 y1 x2 y2). Вторая строка – тоже самое для второй прямой. Третья строка содержит число N - количество точек (0 < N < 1000). Следующие строки содержат координаты точек (два числа через пробел). | 0 – нет;  1 – да, все точки находятся между прямыми. |
| 18 | Заданы координаты N точек. Образуют ли эти точки выпуклый многоугольник? | Первая строка содержит число N (2 < N < 1000). Следующие строки содержат координаты точек (два числа через пробел). | 0 – нет, не образуют;  1 – да, образуют. |
| 19 | Выполнить операцию деления многочлена n-ой степени  на многочлен первой степени (x-c) используя схему Горнера. | Первая строка – число n (степень многочлена). Вторая строка – коэффициенты a0, a1, …, an. Т.е. n+1 число через пробел. Третья строка – число c. | Формат выходного файла аналогичен формату исходных данных. Файл должен содержать три строки: степень многочлена-частного, его коэффициенты, остаток от деления. |
| 20 | Заданы две матрицы. Требуется вычислить их произведение. | Первая строка – размерность первой матрицы (число строк и число столбцов через пробел; в сумме не более 1000). Далее – сама матрица, записанная построчно (в строке числа разделены пробелом). Следующая строка – размерность второй матрицы. Далее – вторая матрица (построчно). | Формат выходного файла аналогичен формату входных данных. Первая строка – размерность матрицы-произведения. Далее – сама матрица построчно. |
| 21 | Задана квадратная матрица. Требуется вычислить её определитель. | Первая строка – размерность матрицы (не более 20). Далее – сама матрица, записанная построчно (в строке числа разделены пробелом). | Определитель. |
| 22 | Заданы N трехмерных векторов. Требуется определить их векторное произведение. | Первая строка содержит число N (1 < N < 1000). Следующие строки содержат координаты векторов (три числа через пробел: x y z). | Вектор [a, b]. Одна строка с его координатами через пробел. |
| 23 | Заданы три вектора. Найти смешанное (скалярно-векторное) произведение этих векторов. | Три строки. Каждая строка содержит координаты одного из векторов (три числа через пробел: x y z). | Число <a, b, c>. Одна строка с единственным числом. |
| 24 | Найти седловые точки матрицы. Матрица А имеет седловую точку Аi,j, если Аi,j является минимальным элементом в i-й строке и максимальным в j-м столбце. | Первая строка – размерность матрицы (число строк и число столбцов через пробел; в сумме не более 1000). Далее – сама матрица, записанная построчно (в строке числа разделены пробелом). | Каждая строка – координаты (строка и столбец) найденной седловой точки. Координаты записываются через пробел, номера строк/столбцов считаются начиная с 1. Если седловых точек нет – создать пустой файл. |
| 25 | Дан некоторый текст. Найти слово (слова), которое (которые) наиболее часто встречается в этом тексте (без учёта регистра букв). | Текстовый файл с произвольным текстом, не более 1М символов. | Слово, которое больше всего раз повторяется в тексте. Если таких слов несколько, то каждое пишется в отдельную строку. |
| 26 | Зашифровать заданный текст. | Первый файл - текстовый файл с произвольным текстом, не более 1М символов. Второй файл – корректный ключ для шифрования. В нем содержится две строки символов (один над одним), разделенных пробелами. Шифрование производится заменой верхнего символа нижним. | Зашифрованный файл (с сохранением строчной структуры исходного файла). |
| 27 | Решение уравнения Ax2 + Bx + C = 0 | Одна строка с коэффициентами A, B, C, разделенными пробелами | Строка с решением уравнения (одно или два различных числа), либо строка “No solutions”, либо строка “Infinity of solutions” |
| 28 | Решение системы уравнений  A1x + B1y = C1  A2x + B2y = C2 | Две строки с коэффициентами A1, B1, C1 и A2, B2, C2, разделенными пробелами | Строка с решением системы (два числа- x и y), либо строка “No solutions”, либо строка “Infinity of solutions” |