

# Задачи темы «Многопоточность»

## Требования к выполнению задач

Реализовать в графическом интерфейсе пользователя или в командной строке. Для командной строки, программа завершается по нажатию сочетания клавиш `ctrl+c`. Все дополнительные потоки программы должны завершиться корректно перед завершением основного потока. При завершении потока должно быть выведено сообщение о его остановке. При выводе данных не должно быть разрывов. При использовании условных переменных всегда должна быть проверка на ложное пробуждение. Для вывода текста разрешено использовать только `cout`.

## Задачи

1	Напишите программу, которая считывает натуральное число $n$ ( $n > 3$ ), затем создаёт $n$ потоков (каждый из потоков получает номер, начиная с 1). Все дополнительные потоки после старта переходят в режиме ожидания. Основной поток раз в несколько секунд выводит из режима ожидания случайный поток, после чего тот печатает свой номер и переходит обратно в режим ожидания.	4
2	Используя <code>future/promise</code> , напишите программу для поиска индекса наименьшего и наибольшего элемента в массиве при помощи асинхронных вызовов функции (далее, задач) поиска на интервале внутри массива (поиск на одном интервал равен одной задаче). Массив состоит из случайных целочисленных значений (количество элементов в массиве более 100 000, значение одного элемента в интервале от -2147483648 до 2147483647). Количество элементов задаётся пользователем. Для одного массива измерьте время выполнения поиска для 2, 4, 8, 16 задач. Для каждого количества задач выведите затраченное на выполнение задачи время.	6
3	Напишите программу, которая генерирует две матрицы со случайными целочисленными элементами, затем вычисляет произведение первой и второй матрицы. Размеры матриц задаются пользователем. Минимально допустимое количество строк и столбцов в результирующей матрице - 50. Генерация матриц должна выполняться в двух или более потоках параллельно. Вычисление произведения должно выполняться параллельно в двух или более потоках. Выполните проверку корректности вычисления.	9
4	Напишите программу, которая моделирует систему массового обслуживания с накопителем конечной ёмкости. Одному прибору соответствует один поток. После извлечения заявки из очереди, поток прибора засыпает на случайное время. После окончания этого времени, прибор считается свободным. Ёмкость накопителя и количество приборов (всегда больше двух) задаётся пользователем. Генератор через случайные промежутки времени добавляет заявки в очередь. Генератор неактивен, если в очереди	15

	нет свободных мест. Генератор работает в отдельном потоке. Во время работы программы на экран выводится информация о состоянии приборов (занят (сколько времени осталось до завершения обработки), свободен) и количестве элементов в очереди.	
5	Напишите программу, которая моделирует систему массового обслуживания с накопителем конечной ёмкости. Приборы разделены на $n$ ( $n > 2$ ) групп, одна группа приборов обслуживает один класс заявок. Внутри класса заявки разделены на три типа: заявки третьего типа имеют наивысший приоритет, заявки первого типа - наименьший. В каждой группе $m$ ( $m > 2$ ) приборов. Все типы заявок находятся в общей очереди. Работа одного прибора происходит в отдельном потоке. После извлечения заявки из очереди, поток прибора засыпает на случайное время. После окончания этого времени, прибор считается свободным. Ёмкость накопителя, количество групп приборов, количество приборов в каждой группе задаётся пользователем. Генератор через случайные промежутки времени добавляет в очередь заявки из разных групп со случайным типов. Генератор неактивен, если в очереди нет свободных мест. Генератор работает в отдельном потоке. Во время работы программы на экран выводится информация о состоянии приборов (занят (каким типом и сколько времени осталось до завершения обработки), свободен) и количестве элементов в очереди.	21,5