

## Задания к лабораторной работе №2.

Написать программу для консольного процесса, который состоит из трёх потоков: **main**, **work**, и третьего (см. варианты).

*Глобальные переменные не использовать!*

### Индивидуальные варианты:

3.1 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива НА КОНСОЛЬ; ввести число *k*;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будут выведены на консоль *k* элементов массива).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком main, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Найти в массиве неповторяющиеся элементы (разместить их в массиве слева, остальные соответственно справа). Элементы - символы.  
известить поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком main, использовать бинарный семафор!*):

ждёт от потока **main** сигнал о начале суммирования;  
выполнить суммирование элементов итогового массива до заданной позиции *k*; вывести итоговую сумму.

3.2 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу используя бинарный семафор для синхронизации с потоком main, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Найти в массиве повторяющиеся элементы (разместить их группы в массиве слева, остальные соответственно справа). Элементы - вещественные числа.  
известить поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будет сформирован итоговый массив).

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком work, использовать критическую секцию!*):

ждёт от потока **work** сигнал о начале суммирования;  
выполнить суммирование элементов итогового массива; вывести итоговую сумму.

### 3.3 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли;  
вывести размерность и **ЭЛЕМЕНТЫ ИСХОДНОГО МАССИВА НА КОНСОЛЬ**;  
ввести число **k**;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток **stdout** после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) **параллельно с работой потока work**;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будут выведены на консоль **k** элементов).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком **main** – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком **main**, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Сортировка методом “пузырька”. Элементы - вещественные числа двойной точности.  
известить поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком **main**, использовать бинарный семафор!*):

ждёт от потока **main** сигнал о начале суммирования;  
выполнить суммирование элементов итогового массива до заданной позиции **k**; вывести итоговую сумму.

### 3.4 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
ввести число **k**;  
запустить поток **work**;  
запустить поток **MultElement**;  
освобождение выходной поток **stdout** после **ВЫВОДА НА КОНСОЛЬ КАЖДОГО НОВОГО ЭЛЕМЕНТА МАССИВА**.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком **main** – использовать семафор. Проверить работу используя бинарный семафор для синхронизации с потоком **main**, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве элементов из диапазона [A,B] (разместить их в массиве слева, остальные элементы массива - заполнить нулями). Элементы - целые числа без знака. Числа A,B ввести в потоке **main**.  
известить поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток **MultElement** о начале работы (момент запуска произойдёт после того, будет сформирована часть итогового массива (когда будут найдены все элементы из диапазона [A, B])).

Поток **MultElement** должен выполнить следующие действия (*Для синхронизации с потоком **work**, использовать критическую секцию!*):

ждёт от потока **work** сигнал о начале работы;  
выполнить произведение элементов итогового массива (когда будут найдены все элементы из диапазона [A, B]);  
вывести произведение.

### 3.5 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли;

вывести размерность и элементы исходного массива на консоль; ввести число *k*;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.

выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будут выведены на консоль *k* элементов массива).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком main, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Сортировка выбором. Элементы - символы.  
извещать поток `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком main, использовать бинарный семафор!*):

ждёт от потока `main` сигнал о начале суммирования;  
выполнить суммирование элементов (кодов СИМВОЛОВ) итогового массива до заданной ПОЗИЦИИ *k*;  
вывести итоговую сумму.

3.6 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
ввести число *k*;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива. Запросить число *A*.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу используя бинарный семафор для синхронизации с потоком main, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве элементов  $>A$  (разместить их в массиве слева, остальные элементы массива - заполнить нулями). Элементы - целые числа без знака. Число *A* ввести в потоке `main`.  
извещать поток `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будет сформирован итоговый массив).

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком work, использовать критическую секцию!*):

ждёт от потока **work** сигнал о начале суммирования;  
выполнить суммирование элементов итогового массива; вывести итоговую сумму.

3.7 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
ввести число *k*;  
запустить поток **work**;  
запустить поток **SumElement**;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.

выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;  
известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будут готовы к печати  $k$  - элементов массива).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком **main** – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком **main**, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве простых чисел (разместить их в массиве слева, остальные элементы массива - справа).  
Элементы - целые числа без знака.  
извещать поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком **main**, использовать бинарный семафор!*):

ждёт от потока **main** сигнал о начале суммирования;  
выполнить суммирование элементов итогового массива до заданной позиции  $k$ ; вывести итоговую сумму.

3.8 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли;  
вывести размерность и элементы исходного массива на консоль; запустить поток **work**;  
запустить поток **CountElement**;  
освобождение выходной поток **stdout** после вывода на консоль каждого нового элемента массива.  
Запросить символ  $X$ .  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком **main** – использовать семафор. Проверить работу программы используя бинарный семафор для синхронизации с потоком **main**, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве элементов  $=X$  (разместить их в массиве слева, остальные элементы массива - справа). Элементы - символы.  $X$  ввести в потоке **main**.  
извещать поток **main** о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток **CountElement** о начале работы (момент запуска произойдёт после того, будет сформирован итоговый массив).

Поток **CountElement** должен выполнить следующие действия (*Для синхронизации с потоком **work**, использовать критическую секцию!*):

ждёт от потока **work** сигнал о начале суммирования;  
подсчитать количество элементов равных  $X$ ;  
вывести итоговую сумму.

3.9 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
ввести число  $k$ ;  
запустить ПОТОК **work**;  
запустить поток **MultElement**;  
освобождение выходной поток **stdout** после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;  
известить поток **MultElement** о начале работы (момент запуска произойдёт после того, будут выведены на консоль  $k$  элементов массива).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком **main** – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком **main**, объяснить отличия, если есть!*):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;

Поиск в массиве элементов  $<A$  (разместить их в массиве слева, остальные элементы массива - справа). Элементы - вещественные числа. Число  $A$  ввести в потоке `main`.  
извещать поток `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;

Поток `MultiElement` должен выполнить следующие действия (Для синхронизации с потоком `main`, использовать бинарный семафор!):

ждёт от потока `main` сигнал о начале суммирования;  
выполнить произведение элементов итогового массива до заданной позиции  $k$ ; вывести итоговое произведение.

3.10 Поток `main` должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
запустить поток `work`;  
запустить поток `SumElement`;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока `work`;

Поток `work` должен выполнить следующие действия (Для синхронизации с потоком `main` – использовать семафор. Проверить работу используя бинарный семафор для синхронизации с потоком `main`, объяснить отличия, если есть!):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве лексем, (разделители – цифры). Полученные лексемы поместить в массиве слева, разделитель - пробел, остальные элементы - заполнить символом '0'. Элементы массива - символы.  
извещать поток `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток `SumElement` о начале суммирования (момент запуска произойдёт после того, будет сформирован итоговый массив.

Поток `SumElement` должен выполнить следующие действия (Для синхронизации с потоком `work`, использовать критическую секцию!):

ждёт от потока `work` сигнал о начале суммирования;  
выполнить суммирование элементов (кодов) итогового массива; вывести итоговую сумму.

3.11 Поток `main` должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с КОНСОЛИ; вывести размерность и элементы исходного массива на консоль;  
ввести число  $k$ ;  
запустить поток `work`;  
запустить поток `SumElement`;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой ПОТОКА `work`;  
известить поток `SumElement` о начале суммирования (момент запуска произойдёт после того, будут выведены на консоль  $k$  элементов массива).

Поток `work` должен выполнить следующие действия (Для синхронизации с потоком `main` – использовать семафор. Проверить работу используя бинарный семафор для синхронизации с потоком `main`, объяснить отличия, если есть!):

запросить у пользователя временной интервал, требуемый для отдыха после ПОДГОТОВКИ одного элемента в массиве;  
Приведение массива к палиндрому (получившейся палиндром поместить в массиве слева, а лишние элементы соответственно – справа ) Элементы - символы  
извещать поток `main` о новом элементе;

после каждого готового элемента ОТДЫХАТЬ в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком main, использовать критическую секцию!*):

- ждёт от потока **main** сигнал о начале суммирования;
- выполнить суммирование элементов (кодов) итогового массива до заданной позиции **k**;
- вывести итоговую сумму.

3.12 Поток **main** должен выполнить следующие действия:

- создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива НА КОНСОЛЬ;
- запустить поток **work**;
- запустить поток **MultElement**;
- освобождение выходной поток **stdout** после вывода на консоль каждого нового элемента массива.
- выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;

ПОТОК **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком main, объяснить отличия, если есть!*):

- запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;
- Сортировка выбором. Элементы - целые числа.
- извещать поток **main** о новом элементе;
- после каждого готового элемента ОТДЫХАТЬ в течение заданного интервала времени;
- известить ПОТОК **MultElement** о начале работы (момент запуска произойдёт после того, будет сформирован итоговый массив).

Поток **MultElement** должен выполнить следующие действия (*Для синхронизации с потоком work, использовать бинарный семафор!*):

- ждёт от потока **work** сообщения о начале суммирования;
- выполнить произведение элементов итогового массива; вывести произведение.

3.13 Поток **main** должен выполнить следующие действия:

- создать массив, размерность и элементы которого вводятся пользователем с консоли;
- вывести размерность и ЭЛЕМЕНТЫ ИСХОДНОГО МАССИВА НА КОНСОЛЬ;
- ввести число **k**;
- запустить поток **work**;
- запустить поток **SumElement**;
- освобождение выходной поток **stdout** после вывода на консоль каждого нового элемента массива.
- выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока **work**;
- известить поток **SumElement** о начале суммирования (момент запуска произойдёт после того, будут выведены на консоль **k** элементов массива).

Поток **work** должен выполнить следующие действия (*Для синхронизации с потоком main – использовать семафор. Проверить работу программы используя критическую секцию для синхронизации с потоком main, объяснить отличия, если есть!*):

- запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;
- Поиск в массиве элементов, соответствующих цифрам (слева поместить в массив цифры, а остальные элементы массива - заполнить пробелами). Элементы - символы.
- извещать поток **main** о новом элементе;
- после каждого готового элемента ОТДЫХАТЬ в течение заданного интервала времени;

Поток **SumElement** должен выполнить следующие действия (*Для синхронизации с потоком main, использовать бинарный семафор!*):

- ждёт от потока **main** сообщения о начале суммирования;
- выполнить суммирование элементов (кодов) итогового массива ДО ЗАДАННОЙ ПОЗИЦИИ **k**; вывести итоговую сумму.

3.14 Поток **main** должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
запустить ПОТОК `work`;  
запустить поток `SumElement`;  
освобождение выходной поток `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока `work`;

Поток `work` должен выполнить следующие действия (Для синхронизации с потоком `main` – использовать **семафор. Проверить работу используя бинарный семафор для синхронизации с потоком `main`, объяснить отличия, если есть!**):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве лексем, начинающихся с цифры (разделители – пробел и тире). Полученные лексемы поместить в массиве слева, а лишние элементы - заполнить символом подчеркивания: «\_»). Элементы - символы.  
извещать ПОТОК `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток `SumElement` о начале суммирования (момент запуска произойдет после того, будет сформирован итоговый массив.

Поток `SumElement` должен выполнить следующие действия (Для синхронизации с потоком `work`, использовать критическую секцию!):

ждёт от потока `work` сообщения о начале суммирования;  
выполнить суммирование элементов (кодов) итогового массива; вывести итоговую сумму.

3.15 Поток `main` должен выполнить следующие действия:

создать массив, размерность и элементы которого вводятся пользователем с консоли; вывести размерность и элементы исходного массива на консоль;  
запустить поток `work`;  
запустить поток `Sum/CountElement`;  
освобождение выходной ПОТОК `stdout` после вывода на консоль каждого нового элемента массива.  
выводить на экран поэлементно элементы массива (итогового) параллельно с работой потока `work`;

Поток `work` должен выполнить следующие действия (Для синхронизации с потоком `main` – использовать **семафор. Проверить работу используя бинарный семафор для синхронизации с потоком `main`, объяснить отличия, если есть!**):

запросить у пользователя временной интервал, требуемый для отдыха после подготовки одного элемента в массиве;  
Поиск в массиве лексем, начинающихся с цифры (разделители – пробел и тире). Полученные лексемы поместить в массиве слева, а лишние элементы - заполнить символом подчеркивания: «\_»). Элементы - символы.  
извещать поток `main` о новом элементе;  
после каждого готового элемента отдыхать в течение заданного интервала времени;  
известить поток `Sum/CountElement` о начале суммирования (момент запуска произойдет после того, будет сформирован итоговый массив.

Поток `Sum/CountElement` должен выполнить следующие действия (Для синхронизации с потоком `work`, использовать критическую секцию!):

ждёт от потока `work` сообщения о начале суммирования;  
выполнить суммирование и подсчёт элементов (до символов подчеркивания: «\_») итогового массива,; вывести результаты.

