**Лабораторная №6**

**ТРИ ТОЛСТЯКА**

****

***Условие:***

Толстяки очень прожорливые, и готовы кушать наггетсы без перерыва.   
У них есть повар Кук, которого они наняли, чтобы готовил еду круглосуточно. На момент, когда его наняли, на столе было 3 тарелки с 3000 наггетсов в каждой ( int dish1 = 3000 (dish2, dish3) ). У каждого толстяка, своя тарелка.

Если наггетсы закончатся хотя бы у одного толстяка - повара придется уволить.

Скушав больше 10 000 наггетсов толстяк самоуничтожается.

Если лопнут все 3 толстяка, Кук не получит зарплату.

По истечению 5 дней, у повара закончаться силы, и он не выдержит и решит уволится сам!!!

***Задача:***

Написать программу на С++, где каждый толстяк в своём потоке жрёт наггетсы со своей тарелки. Повар в своём потоке эти наггетсы готовит и выкладывает на стол.

У каждого толстяка есть коэффициент обжорства - gluttony : столько наггетсов он съедает за раз.

например - const int gluttony = 3;

У Кука есть коэффициент производительности - efficiency factor : столько за раз (за одну итерацию в цикле) наггетсов выкладывается на стол, раскладывая по 3 тарелкам сразу.

например - const int efficiency\_factor = 3;   
Когда Кук выкладывает наггетсы на стол, толстяки ждут, чтобы случайно не откусить ему руки. ( тут про мьютексы идет речь )

1 день = 1 секунда компа.

При каких коэффициентах прожорливости толстяков и производительности Кука выполняться условия? (вывести в терминал):

1. Кука уволили
2. Кук не получил зарплату.
3. Кук уволился сам!

Подобрать значения efficiency\_factor и gluttony, при которых выполняются те или иные условия.

Не смотря на блокировку через мьютекс, толстяки могут есть долго и не давать куку выложить наггетсы в тарелку. Или наоборот Кук заблокировал тарелки, выкладывает наггетсы, а толстяки каждый раз обращаясь к тарелкам не успевают их взять, а Кук продолжает выкладывать. **Такая проблема называется *ресурсным голоданием*, когда поток не может захватить долгое время критический ресурс, хотя казалось бы логика программы это позволяет.**

Выкладку и поедание наггетсов надо синхронизировать: Кук выложил по 1 на тарелку, потом толстяки съели по 1 шт. (или больше, в зависимости от коэффициентов прожорливости и производительности). Один из возможных способов можно реализовать синхронизацию потоков с помощью доп. переменных и функции **yield**.

Чтобы не путаться с вызовами lock и unlock, из-за чего можно получить deadlock(), можно обернуть толстяков и Кука в класс, содержащий мьютекс. Так же разрешено использовать любые виды и обертки мьютексов, как [std::lock\_guard](https://en.cppreference.com/w/cpp/thread/lock_guard).