* Создать на языке Си функцию,
  + Умножающую два вектора (векторное умножение – в результате вектор). Размер векторов - **N** элементов типа data\_in. Результат – вектор - **N** элементов типа data\_out.
    - Аргументы функции: два входных вектора, выходной вектор
  + В файле lab4\_1.h должны быть определены: **N = 4**, тип данных data\_in, имеющий тип short; тип данных data\_out, имеющий тип int
  + Дать самому внутреннему циклу, реализованному в функции, имя L1, остальным L2…
* Создать на языке Си тест для проверки работы функции. Тест должен обеспечивать
  + Запуск функции 3 раза
    - Заполнение векторов случайными значениями.
    - Перед следующим запуском функции не забудьте обнулить ожидаемое значение.
  + проверку правильности полученного функцией результата.
* Отладить функцию и тест (при неправильном результате в любом из запусков функции тест должен сообщать об ошибке).
* Создать скрипт автоматизирующий процесс:
  + Создания проекта lab4\_1,
  + Назначения функции lab4\_1
  + Подключения файла lab4\_1.c (папка source ),
  + Подключения файла lab4\_1\_test.c (папка source),
  + Создания решения
    - sol1, для которого
      * Микросхема: xa7a12tcsg325-1Q
      * Период тактового сигнала: 10нс, uncertainty 1нс.
      * Си моделирование
      * ОТКЛЮЧИТЬ конвейеризацию для всех циклов
      * Синтез
      * С/RTL cosimulation (с опцией Dump Trace = Port)
    - sol2, для которого
      * для цикла L1 задать директиву **Pipeline II=1**
      * Синтез
      * С/RTL cosimulation (с опцией Dump Trace = Port)
    - Sol3, для которого
      * для цикла L1 задать директиву **Pipeline II=1**
      * **для внешнего цикла задайте Pipeline off и rewind**
      * Синтез
      * С/RTL cosimulation (с опцией Dump Trace = Port)
    - Sol4, для которого
      * для цикла L1 задать директиву **Pipeline off**
      * **для внешнего цикла задайте Pipeline II=1 и rewind**
      * Синтез
      * С/RTL cosimulation (с опцией Dump Trace = Port)
* Отладить и проверить работу созданного скрипта.
* После выполнения скрипта открыть GUI
* Убедиться, что созданы все решения
* Используя средства HLS проверить, сравнить и зафиксировать использованные интерфейсы для каждого из трех решений
* Используя средства HLS сравнить полученные решения (привести Report Comparison)
  + Timing
  + Latency
  + Utilization Estimation
* Используя средства HLS сравнить и привести результаты планирования (Schedule viewer) для каждого из трех решений

Для каждого из трех решений привести временные диаграммы

* + Решение 1
    - Надо показать
      * II, Latency для первого запуска функции
  + Решение 2
    - Надо показать
      * II, Latency для первого запуска функции
      * Два запуска функции
  + Решение 3
    - Надо показать
      * Три (два) запуска функции
  + Решение 4
    - Надо показать
      * Три (два) запуска функции
* Оформить отчет, который должен включать
  + Задание
  + Раздел с описанием исходного кода функции
  + Раздел с описанием теста
  + Раздел с описание созданного командного файла
  + Раздел с описанием результатов сравнения решений (со снимками экрана из vitis HLS)
  + Раздел с анализом временных диаграмм (со снимками экрана временных диаграмм)
    - Необходимо привести пояснения особенностей работы каждой из реализаций
  + Раздел с анализом результатов по заполненному xls файлу (по оси х – решение, по У - производительность, аппаратные затраты). (со снимками экрана с заполненной таблицей и полученным графиком)
    - Анализ и выбор оптимального (критерий максимальная производительность) решения
  + Выводы
* Архив должен включать всю рабочую папку проекта, отчет и файл с электронной таблицей.