Тестовое задание для кандидатов на роли SD и SDT. Вариант 1-1

# Информация для кандидата

Данное задание предназначено для кандидатов на роли SD (Software Developer) и SDT (Software Developer in Test). Необходимо реализовать его на одном из следующих языков программирования:

* C#
* C/C++.

Использование стандартных библиотек приветствуется, но только для вспомогательных целей: выделение, освобождение ресурсов, хранение данных и т.д. Ключевые алгоритмы, составляющие суть задачи, необходимо реализовать самостоятельно. Если Вы знаете, как реализовать его с помощью уже имеющихся в стандартных библиотеках функций, можете описать это в комментариях – это также приветствуется.

Реализуйте то решение, которое покажется Вам оптимальным. В комментариях обоснуйте Ваше решение.

Написанный Вами программный код должен компилироваться и корректно работать. Кроме этого мы будем оценивать то, насколько аккуратно Вы пишете, как структурируете текст, как комментируете его.

Вы можете реализовать любой из способов взаимодействия с пользователем. Тем не менее, если Ваша программа будет удобна в обращении, это будет дополнительным плюсом.

Спасибо, что уделили внимание нашей Компании. Удачи!

# Описание задачи

На вход дан неупорядоченный набор объектов, каждый из которых помечен одним из трёх цветов: красный, зелёный и синий. Также на вход даётся правило, устанавливающее отношение порядка между цветами. Требуется упорядочить объекты в соответствии с указанным отношением порядка цветов.

**Пример:**

(Обозначения: **К**, **З**, **С** – объекты, помеченные красным, зелёным и синим цветами, соответственно.)

Входной набор объектов:   
**С С** **З** **С К З** **З** **З** **К К С З С С К З**

Задано следующее отношение порядка цветов:   
**З < С < К**

На выходе должен быть следующий набор объектов:  
 **З З** **З** **З З** **З С С С С С С К К К К**

Необходимо реализовать описанный алгоритм в виде функции (или метода класса) и написать код, тестирующий его.

# Требования к реализации

Задача должна быть решена с использование двух потоков: генерирующего и обрабатывающего. Генерирующий поток формирует исходные данные, т.е. неупорядоченные последовательности объектов. Обрабатывающий поток упорядочивает сформированные последовательности в соответствии с указанным алгоритмом.

Начало и прекращение работы потоков должно происходить по команде пользователя.

Дополнительные требования к реализации:

- CMake;

- boost.thread или std.thread для реализации потоков/синхронизации.