Cartesian Vectors

In this exercise, you are asked to implement basic operations on cartesian vectors using operator overloading.

These vectors are configured by config.h, where the type of each element as well as the number of elements per vector are defined.

The usage of the vectors is showcased in main.cc. You need to implement all operators necessary to compile this file.

Note that the operator * between vectors is interpreted as the dot product (therefore returning a single value), whereas the * between a scalar/a value and a vector is interpreted as scaling the vector.

One of the constructors needs to use an initializer list, look here: initializer_list

To structure your implementation, I suggest you go first through main.cc and write down the signatures of all necessary operations.

Restrictions

You are not allowed to use std::array, std::vector, std::list, etc (Anything that is already a container)

Hints

std::initializer_list is a somewhat weird construct. You can imagine it like a wrapper around an array constructed from a braced list of values, like {1, 2, 3}. To access the underlying array, you should can std::data.

Challenge

Implementations faster than our (suboptimal but not bad) reference implementation get extra points.

Bonus

Another bonus is given to an implementation if it is at least 10% faster than any of the other implementations (including our reference implem)