

In C++ write the generic expected monad. This is a type wrapper that removes error conditions as a side effect.

Expected<T> contains either a T or the error that prevented it from holding a T. We will use an exception pointer.

Expected<T> is implicitly convertible to T. If it contains an error it throws on conversion.

Expected<T> has an apply function that runs a function on the value returns an expected result.

Given U f(T), and Expected<T> has a value return Expected<U> containing result of f. Otherwise return an expected with the error.

For convenience implement the mixed mode operators (+,-,/,*,%,<,>,<=,>=,! =) in terms of apply.

Write a main function that shows all of these cases.