Goal

The goal of this exercise is to show you the connection between operator overloading and generic programming

You are given two files:

A *main.cc* which you are not allowed to change and which highlights the usage and desired additional features of std::set

A *set_operations.hpp* in which you will code the functionalities. Note that the only allowed includes in *set_operations.hpp* are #include<set>, #include<type_traits> and #include<concepts>.

Additional functionalities

Given the sets *s1* and *s2* (over any [admissible]) type we want and an (positive) integral n:

- *s1* + *s2* to return a new set containing the union of *s1* and *s2*
- *s1* - *s2* to return a new set containing the difference of *s1* and *s2*, that is all values of *s1* not in *s2*
- *s1* ^ *s2* to return a new set containing the symmetric difference of *s1* and *s2* (*xor* of *s1* and *s2*)
- *s1* * *s2* to return a new set {x1+x2 | forall x1 in s1 and x2 in s2}
- *s1* ^ *n* to return a new set containing the original set multiplied with itself n times