

Задача 1.

Да се реализира клас `Sorted<T, capacity>`, който реализира "винаги сортиран" масив от елементи `T`. В `Sorted` може да има максимум `capacity` елемента от тип `T`. Класът да има методи `add(T& elem)`, `remove (T& elem)` и `print()`, които съответно добавят елемент, махат елемент и отпечатват елементите разделени с нов ред.

Задача 2. Да се дефинира шаблон на клас `Relation<T>`, който съдържа два обекта от тип `T`, наречени `subject` и `object`, и низ с произволна дължина `relation`, описващ връзката между тези обекти.

Пример: `Relation<int> r1(2,6,"is smaller than"),r2(6,3,"is divisible by");`

За шаблона да се реализират голямата четворка и операция за отпечатване `void print()`

Пример: `r1.print(): 2 is smaller than 6.`

За инстанцията на шаблона `Relation<int>` реализирайте и оператор за композиция `*` по следния начин. Ако `r = r1 * r2`, то `r.subject = r1.subject`, `r.object = r2.object`, `r.action = r1.action r1.object “, which is“ r2.action`

Пример: `(r1*r2).print(): 2 is smaller than 6, which is divisible by 3` Композицията се допуска само ако `r1.object == r2.subject`, в противен случай резултатът е `r1`.