Проект 4. Мултисет (Multiset)

В рамките на този проект трябва да разработите собствена структура от данни Мултисет. Мултисет е структура от данни, подобна на множество, но може да запазва много елементи с еднакви стойности.
За тази структура трябв да реализирате следните функции:

- add - Добавя нов елемент в нея.
- remove - Премахва конкретен елемент по ключ. В случай на повече от един елемент с този ключ, премахва само един еквивалентен на подадените данни.
- removeAll - Премахва всички елементи с подаден ключ.
- isSubSet- Проверява дали даден мултисет е подмножество на друг мултисет. Реализирайте тази операция като функция, но също и чрез предефинирани оператори.
- count с аргумент - Връща броя на срещанията на елементи с подадения ключ. Реализирайте и подходящ индексиращ оператор с подобна функционалност.
- count без аргумент - Връща броя на всички елементи в мултисета.
- merge - Слива две мултимножества в ново мултимножество.
- Вашите данни трябва да могат да преживеят рестартиране на системата. Например можете да ги записвате в общ файл или да поддържате различен файл за всеки блок.

Сами можете да изберете типа на функциите и на техните параметри, а също така трябва да изберете какво да бъде тяхното поведение при възникване на грешка (например ако на remove бъде подаден невалиден ключ). Можете сами да определите какви са изискванията към типа на ключовете, но вашата структура трябва да работи безпроблемно със стандартните скаларни типове.
Възможно е да реализирате съхранението в паметта, но трябва да гарантирате, че при внезапно спиране на тока, данните няма да бъдат повредени и да има минимални загуби. Разбираемо е, че ако токът спре по време на запис, данните от текущата операция може да не бъдат съхранени. Същевременно, незавършилата операция не трябва да поврежда хранилището.

Пример:

```
Multiset<int> mSet;
mSet.add(1);
mSet.add(2);
mSet.add(1);
mSet.add(3);

std::cout<<mSet.count(1)<<std::endl; // извежда 2
std::cout<<mSet.count()<<std::endl; // извежда 3


Multiset<int> anotherSet;
anoterSet.add(2);
anotherSet.add(2);
anotherSet.add(1);

std::cout<<anotherSet.isSubSet(mSet); // true
std::cout<<mSet.isSubSet(anotherSet); // false
```