

Да се имплементира клас за дата и час(обичайни имена са `Date`, `Time` или `DateTime`). Класът трябва да има числови атрибути за секунди, минути, часове, ден, месец и година. Веднъж инициализирани атрибутите не могат да се променят. Добавете конструктори:

- по подразбиране - всички атрибути се инициализират с 0
- с аргументи за всички атрибути - атрибутите се инициализират чрез инициализиращ списък
- копиращ

Към класа добавете константен статичен член за нулева дата. Това е инстанция на класа за дата, за която са подадени:

- секунди = 0
- минути = 0
- часове = 0
- дни = 0
- месеци = 0
- година = 1900

Дни и месеци се броят от 0!

Това е нулавата(началната) дата и всяка инстанция на класа ще представлява отстояние от нея вместо от пълната нула.

Дата с (година = 100, месеци = 2, дни . 17) представлява (18 март 2000г.) защото $1900 + 100 = 2000$.
--

Предефинирайте операторите:

- `==`, `!=`, `>`, `<`, `>=`, `<=` - сравняват текущата инстанция с друга дата
- `+` - добавя към текущата инстанция всички атрибути на друга дата
- `-` - изважда от текущата инстанция всички атрибути на друга дата
- `<<` - извежда датата във формат `YYYY-MM-DD hh:mm:ss` *Взимайте предвид отстоянието от нулавата дата*

За да имплементирате правилно операторите за сравнение добавете скрит метод, който връща сумата от равностойността в секунди на всички атрибути на инстанцията.

Дефинирайте към класа статичен метод, който получава референция към списък от дати и го сортира.

Конструкторите и операторите да хвърлят `std::invalid_argument` изключения ако са подадени дати с отрицателни стойности.

Демонстрирайте в `main` работата на всичко дефинирано.

Допълнително:

- ползвайте подходящи модификатори за достъп - отвън да може да се достъпва само каквото е нужно да се достъпва
- добавяйте гетъри и сетъри само при нужда
- ползвайте константи когато нещо няма да бъде променено
- ползвайте референции когато се подават инстанции на класове за да се намали излишното копиране