

Задача 1.

Да се дефинира шаблон на функция, която да намира най-малкия елемент в масив, елементи на който са от произволен тип.

Да се дефинира структура `Point`, представляваща точка в равнината с цели координати.

Като се използва дефинираният шаблон да се сортират елементите на масив от точки в равнината с цели координати. Точката (a, b) е по-малка от (c, d) , ако $a < c$ или $a == c$, но $b < d$.

Задача 2

Да се дефинира клас **AllDayEvent**, който описва събитие със следните характеристики:

- име на събитието – символен низ, разположен в динамичната памет;
- дата – низ с дължина точно 8 символа. Форматът на датата е ггггммдд. Валидирането на стойностите може да се пропусне.

Да се предефинира оператор `<` за клас `AllDayEvent`, така че да проверява дали дадено събитие предшества друго.

Да се дефинира клас **Event**, който наследява `AllDayEvent` и добавя към наследените характеристики следните:

- начален и краен час на събитието. Началният и крайният час са низове с дължина точно 4, символите на които означават час и минути. Форматът е ччмм;
- местоположение – символен низ в динамичната памет.

Да се дефинират подходящи конструктори, селектори и мутатори за всеки от класовете.

Нека множество от събития от тип `AllDayEvent` или `Event` се разглежда като календар. Календарът е представен като масив. Да се избере подходящ тип за елементите на масива.

Да се дефинират следните функции:

- `void filterEvents([подходящ тип] calendar, ..., const char* location)`, която получава като аргументи календар и местоположение и извежда всички събития, включени в календара, които се провеждат на указаното местоположение;
- `bool isSorted([подходящ тип] calendar, ...)`, която проверява дали събитията, включени в календара са последователни.

Забележка: Функциите могат да имат допълнителни параметри, извън описаните.