**Условие**

За да елиминира ръчното броене на бюлетини и неработещите машини за гласуване, държавата е поръчала на вас да направите софтуер, който да брои и сумира автоматично гласовете на всяка партия от всички секции в страната и чужбина.

(За целите на задачата, приемаме, че в страната има само три партии с имената PARTY1, PARTY2, PARTY3.)

Новите машини за гласуване ще съхраняват бройките получени гласове в текстов файл със следния формат:

{SECTION1\_PARTY1\_VOTES} {SECTION1\_PARTY2\_VOTES} {SECTION1\_PARTY3\_VOTES}

{SECTION2\_PARTY1\_VOTES} {SECTION2\_PARTY2\_VOTES} {SECTION2\_PARTY3\_VOTES}

{SECTION3\_PARTY1\_VOTES} {SECTION3\_PARTY2\_VOTES} {SECTION3\_PARTY3\_VOTES}

{SECTION4\_PARTY1\_VOTES} {SECTION4\_PARTY2\_VOTES} {SECTION4\_PARTY3\_VOTES}

...

т.е. един **ред** пази гласовете, получени в една изборна **секция**, като на всеки ред с **един** **интервал** са отделени цели числа, представляващи бройката гласове за съответната партия в тази секция (по 3 цели числа на ред, понеже имаме 3 партии).

Пример за тесктови данни във файл с такъв формат:

5 1 2

6 10 12

20 24 8

14 15 18

5 10 0

Тези данни ни казват, че в секция първа има постъпили 5 гласа за PARTY1, 1 глас за PARTY2 и 2 гласа за PARTY3; в секция втора има по 6, 10 и 12 гласа за съответните партии и т.н. Общо секциите, от които има информация в този файл са 5 на брой.

От вас се изисква да създадете два класа - `SectionVotes` и `ElectionResultsDatabase`.

*SectionVotes****:***

Класът трябва да пази информация за подадените гласове за всяка от партиите в **една** изборна секция. Трябва да съдържа:

* Конструктор с 3 параметъра - гласовете в тази секция за PARTY1, PARTY2 и PARTY3 съответно
* int votesForParty(Party) const - връща колко гласове е събрала дадената партия в тази секция

За да може тази информация да бъде лесно прочетена или записана във файл **с формат като горепосочения**, трябва да бъдат предефинирани операторите `<<` и `>>` за работа със съответните потоци.

*ElectionResultsDatabase:*

Класът трябва да борави с файлове с формат като горепосочения и да пази информация за всички постъпили гласове във всички секции. Има следните методи:

* void addResultsFromFile(const char\* filename): Прочита информация, съдържаща се във файл с име `filename` и с формат като горепосочения.
  + **Не трябва да се трие вече съществуващата информация в класа, а само да се добавя към нея**
  + Съдържанието на файла не трябва да бъде променяно по никакъв начин
  + Очаквайте файловете да бъдат само с коректно форматирани данни
* int votesForParty(Party) const: Връща колко гласове общо е събрала дадената партия
* Party winningParty() const: Връща партията с най-много гласове от изборите.
  + Ако PartyX и PartyY (X < Y) имат еднакъв брой гласове, то в този случай се очаква да върне PartyX
* int numberOfSections() const: Връща от колко изборни секции има информация за гласуването

За лесната работа с този тип файлове, чийто формат е посочен по-горе в условието, трябва и да се предифинират операторите `<<` и `>>` за работа със съответните потоци:

* Операторът за четене от поток не трябва да изтрива съществуващите данни в класа, а само добавя прочетените такива от целия поток.
* Операторът за писане в поток трябва да копира в потока абсолютно всички данни, пазещи се в класа, във формат като горепосочения.