

Задача 1. Програма за съхранение и преглед на записи с оценки на студенти по програмиране (основа)

Да се напише програма, предоставяща възможност за създаване, записване в постоянната памет (във вид на файл във файловата система) и преглед на записи с оценки на студенти. Основно изискване е записаната информация да се съхранява за постоянно и да е достъпна след изход от програмата и последващото и стартиране.

При стартиране програмата очаква на стандартния вход една от следните команди: **create**, **sequentialSearch**, **exit**. Значението на всяка една от тези команди е следното:

- **create** – след въвеждане на тази команда от потребителя се очаква да въведе информация за оценка на студент по програмиране в следния формат: *FN FirstName LastName Grade*, където:
 - *FN* е факултетен номер на студент (цяло положително число с максимална стойност 2^{50});
 - *FirstName* е първото име на студента (низ, с максимална дължина 2^8 символа);
 - *LastName* е фамилия на студента (низ, с максимална дължина 2^8 символа);
 - *Grade* е оценка на студента (цяло положително число в интервала $[2; 61]$);

След въвеждане на информацията и натискане на Enter, програмата автоматично записва въведената информация във файл с името **StudentsGrades.db**. В случай, че файлът **StudentsGrades.db** не съществува, програмата трябва да го създаде автоматично и след това да запише информацията в него. В случай, че файлът **StudentsGrades.db** вече съществува, програмата трябва да го отвори и да допълни съдържанието му с въведената информация.

След записване на информацията, програмата извежда съобщението „**Record saved!**“ и се връща в начало състояние (очаква от потребителя да въведе една от командите: **create**, **sequentialSearch**, **exit**).

- **sequentialSearch** – след въвеждане на тази команда от потребителя се очаква да въведе *FN*. След въвеждане на *FN*, програмата извършва последователно търсене във файла и в случай на точно съвпадение с въведения *FN* на стандартния изход се извежда пълната информация за студента във следния формат *FN FirstName LastName Grade*. В случай, че

запис за студент с въведения *FN* не е намерен, то на стандартния изход се извежда съобщението: „**Record not found!**“. След извеждане на информацията, програмата се връща в начало състояние (очаква от потребителя да въведе една от командите: **create**, **sequentialSearch**, **exit**).

- **exit** – след въвеждане на тази команда се излиза от програмата.

Примерен вход и изход:

Примерен вход:	Изход:
create 100000 Angelina Antonova 6 create 105000 Bilyana Gospodinova 6 sequentialSearch 105000 sequentialSearch 500000 exit	Record saved! Record saved! 105000 Bilyana Gospodinova 6 Record not found!

Задача 2. Програма за управление и преглед на вече съществуващи записи с оценки на студенти по програмиране

Да се напише програма, предоставяща възможност за преглед, обновяване и изтриване в постоянната памет на предварително създадени записи с оценки на студенти (вече съществуващи във вид на файл в файловата система).

Ваша задача е да създадете предварително (по избран от Вас начин) текстов файл с име **StudentsGrades.db**, съдържащ поне 100 000 записа, всеки от които в следния формат: *FN FirstName LastName Grade*, където:

- *FN* е факултетен номер на студент (цяло положително число с максимална стойност 2^{50});
- *FirstName* е първото име на студента (низ, с максимална дължина 2^8 символа);
- *LastName* е фамилия на студента (низ, с максимална дължина 2^8 символа);
- *Grade* е оценка на студента (цяло положително число в интервала $[2;6]$);

Всеки запис за студент във файлът трябва да бъде отделен на нов ред. Факултетния номер *FN* за всеки запис за студент трябва да е уникален за файла (във файла не трябва да присъстват записи с дублиращи се стойности за *FN*).

Основно изискване е действията (приложени с описаните по-долу команди за обновяване и изтриване) да се прилагат върху информация в посочения файл за постоянно и да не се губят след изход от програмата и последващото и стартиране.

При стартиране програмата очаква на стандартния вход една от следните команди: **update**, **delete**, **sequentialSearch**, **exit**. Значението на всяка една от тези команди е следното:

- **update** – след въвеждане на тази команда от потребителя се очаква да въведе новата информация за оценка на студент по програмиране в

следния формат *FN Grade*. След въвеждане на факултетен номер, оценка и натискане на Enter:

- В случай на съществуващ запис за студент с посочения *FN*, програмата автоматично обновява въведената оценка *Grade* в файлът с име **StudentsGrades.db**, извежда съобщението „**Record saved!**“ и се връща в начало състояние (очаква от потребителя да въведе една от командите: **update**, **delete**, **sequentialSearch**, **exit**).
- В случай, че не съществува запис за студент с посочения *FN* програмата извежда съобщение „**Record not found!**“ и се връща в начало състояние (очаква от потребителя да въведе една от командите: **update**, **delete**, **sequentialSearch**, **exit**).
- **delete** – след въвеждане на тази команда от потребителя се очаква да въведе *FN*. След въвеждане на факултетен номер и натискане на Enter:
 - В случай на съществуващ запис за студент с посочения *FN*, програмата автоматично изтрива записът за студент с въведения *FN* от файлът с име **StudentsGrades.db**, извежда съобщението „**Record deleted!**“ и се връща в начало състояние (очаква от потребителя да въведе една от командите: **update**, **delete**, **sequentialSearch**, **exit**).
 - В случай, че не съществува запис за студент с посочения *FN* програмата извежда съобщение „**Record not found!**“ и се връща в начало състояние (очаква от потребителя да въведе една от командите: **update**, **delete**, **sequentialSearch**, **exit**).
- **sequentialSearch** – след въвеждане на тази команда от потребителя се очаква да въведе *FN*. След въвеждане на факултетен номер и натискане на Enter:
 - Програмата извършва последователно търсене търсене за време $O(n)$ във файлът с име **StudentsGrades.db** и в случай на точно съвпадение с въведения *FN*, тогава на стандартния изход се извежда пълната информация за студента в следния формат *FN FirstName LastName Grade*.
 - В случай, че не съществува запис за студент с посочения *FN*, програмата извежда съобщение „**Record not found!**“.
 - След извеждане на информацията, програмата се връща в начало състояние (очаква от потребителя да въведе една от командите: **update**, **delete**, **sequentialSearch**, **exit**).

- **exit** – след въвеждане на тази команда се излиза от програмата.

Извадка с първите два записа от примерен файл StudentsGrades.db, съдържащ поне 100 000 записа :

100000 Angelina Antonova 6 105000 Bilyana Gospodinova 5
--

Примерен вход и изход:

Примерен вход:	Изход:
delete 100000 sequentialSearch 100000 update 105000 6 sequentialSearch 105000 exit	Record deleted! Record not found! Record saved! 105000 Bilyana Gospodinova 6