

Домашна работа №1

КН, 2-ри курс, 2-ри поток

СРОК ЗА ПРЕДАВАНЕ:
посочен в moodle

Изисквания за предаване:

- Предаване на домашното в указания срок от всеки студент във вид на .zip архив със следното име: **(номер_на_домашно)_КН_(курс)_(поток)_(група)_(факултетен_номер)**, където:
 - **(номер_на_домашно)** е число, отговарящо на номерът на домашното, съответстващо на решението (например 1);
 - **(поток)** е число, отговарящо на потока Ви (например 2);
 - **(курс)** е число, отговарящо на курс (например 1);
 - **(група)** е число, отговарящо на групата Ви (например 1);
 - **(факултетен_номер)** е число, отговарящо на факултетния Ви номер (например 12345);
- Архивът да съдържа само изходен код (.cpp и .h файлове) с решение, отговарящо на условията на задачите, като файловете изходен код за всяка задача трябва да са разположени в папка с име **(номер_на_задача)**, където **(номер_на_задача)** е номера на задачата, съответстваща на решението;
- Предаване на архива на посоченото място в moodle;
- Домашното да съдържа текстов файл с име github.txt, съдържащ линк към изходния код в GitHub;
- Спазване на форматирането на данните посочено в Примерен вход и изход;
- Да се пишат коментари за пояснение;

Пример за .zip архив за домашно: 1_KN_1_2_1_12345.zip

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 $[2;6]$);

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! |