

Домашна работа №4

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:
 - Текстов файл с име github.txt, съдържащ линк към изходния код в GitHub;
 - 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;
 - Файлове от друг тип (.db, .txt и др.) са допустими единствено в случай, че това се изисква явно в условието на дадена задача;
- Предаване на архива на посоченото място в moodle;
- 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. Двоично дърво за търсене (Binary Search Tree) на записи за студенти

Да се напише реализация на двоично дърво за търсене на записи за студенти. Достатъчно е информацията да се пази единствено в оперативната памет (RAM).

Всеки запис за студент е в следния формат: *FN FirstName LastName*, 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);

При стартиране програмата очаква на стандартния вход една от следните команди: **insert**, **delete**, **find**, **traverseInorder**, **exit**. След изпълнение на всяка една от тях, програмата се връща в начало състояние (отново очаква от потребителя да въведе някоя от командите). Значението на всяка една от командите е следното:

- **insert** – след въвеждане на тази команда от потребителя се очаква да въведе запис за студент в следния формат *FN FirstName LastName*. След въвеждане и натискане на Enter програмата вмъква (в случай на вече съществуващ запис, програмата заменя съществуващия запис с новия запис) въведения запис в двоичното дърво за търсене и извежда съобщението „**Record inserted!**“.
- **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 двоичното дърво за търсене и извежда съобщението „**Record deleted!**“.
 - В случай, че не съществува запис за студент с посочения *FN* програмата извежда съобщение „**Record not found!**“.

- **find** – след въвеждане на тази команда от потребителя се очаква да въведе *FN*. След въвеждане на *FN*, програмата извършва търсене в двоичното дърво за търсене и в случай на точно съвпадение с въведения *FN* на стандартния изход се извежда пълната информация за студента в следния формат *FN FirstName LastName*. В случай, че не съществува запис за студент с посочения *FN* програмата извежда съобщение „Record not found!“.
- **traverseInorder** – след въвеждане на тази команда от потребителя, програмата извършва inorder обхождане в двоичното дърво за търсене и на стандартния изход се извеждат всеки *FN* отделен със „ „.
- **exit** – след въвеждане на тази команда се излиза от програмата.

Примерен вход и изход:

Примерен вход:	Изход:
insert	Record inserted!
250000 Bilyana Gospodinova	Record inserted!
insert	Record inserted!
120000 Antonia Gospodinova	Record inserted!
insert	Record inserted!
370000 Angelina Angelova	Record inserted!
insert	Record inserted!
300000 Desislava Antonova	Record deleted!
insert	120000, 250000, 300000, 330000,
430000 Angelina Antonova	370000, 430000
insert	
330000 Antonia Antonova	
insert	
10100000 Antonia Antonova	
delete	
10100000	
traverseInorder	
exit	