

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2018-2019

- други домаћи задатак -

Опште напомене:

1. Домаћи задатак 2 састоји се од једног програмска проблема. Студенти проблем решавају **самостално**, на програмском језику C++.
2. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
3. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
4. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса.
Примена рекурзије се неће признати као успешно решење проблема.
6. За све недовољно јасне захтеве у задатку, студенти треба да усвоје разумну претпоставку у вези реализације програма. Приликом одбране, демонстраторе треба обавестити која претпоставка је усвојена (или које претпоставке су усвојене) и која су ограничења програма (на пример, максимална димензија низа и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
7. Одбрана другог домаћег задатка ће се обавити у **среду, 28.11.2018. и четвртак, 29.11.2018.** према распореду који ће накнадно бити објављен на сајту предмета.
8. Пре одбране, сви студенти раде тест знања за рачунаром коришћењем система Moodle (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да се пријаве на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико је студент регистрован на систем путем свог налога електронске поште на серверу mail.student.etf.bg.ac.rs.
9. Име датотеке која се предаје мора бити **dz2p1.cpp**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

19.11.2018. године

Са предмета

Задатак 1 – Имплементација структуре података за убрзавање претраге података о филмовима [100 поена]

Нека је дата структура података за претрагу (база) коју чине подаци о најпопуларнијим филмовима из базе *The Movie Database* (TMDb). Сваки филм је описан записом који чине **наслов** (стринг), **датум издавања** (у формату YYYY-MM-DD, где YYYY представља годину издавања, MM представља месец издавања, а DD представља дан издавања), **језик** (стринг од два знака), **трајање** (у минутима) и **зарада** (реалан број). База за претрагу се састоји од динамичког низа записа које чувају податке о филмовима. База је сортирана неоппадајуће по датуму издавања филма. База је статична и након иницијалног стварања број филмова у њој се неће мењати. Брисање записа о појединачном филму из базе је могуће, али се оно спроводи логички (маркирањем). Стога је у сваки запис о филму потребно додати једно поље које говори да ли је запис валидан или није.

Како би се олакшало и убрзало претраживање овакве структуре података према датуму издавања филма, потребно је креирати **редак индекс коришћењем Б+ стабла произвољног реда** (највише 10). У једном чвору Б+ стабла се чувају **целобројни кључеви који представљају конкатенацију поља YYYY и MM** из датума издавања филма који се налази у бази података. За сваки кључ је везана адреса првог записа са подацима о филму који је издат одговарајуће године и месеца, а логички је валидан у бази.

Реализовати следеће операције:

1. **[20 поена]** стварање и основна манипулација базом података која садржи податке о филмовима
 - стварање на основу података из задате датотеке или са стандардног улаза уз одржавање сортираног поретка записа о филмовима
 - логичко брисање записа о филму из базе (маркирањем)
 - уништавање базе података
 - исписивање садржаја структуре у излазни ток (оператор <<)
2. **[30 поена]** стварање и основна манипулација ретким индексом над базом података која садржи податке о филмовима
 - стварање објекта Б+ стабла **задатог реда** (највише 10)
 - уништавање објекта стабла
 - проналажење кључа у стаблу (враћа вредност `true` ако кључ постоји, односно `false` у супротном)
 - исписивање садржаја стабла у излазни ток (оператор <<), по нивоима, тако да су поједини чворови уочљиви (међусобно размакнути)
3. **[30 поена]** динамичка манипулација ретким индексом
 - уметање кључа у стабло (враћа вредност `false` ако кључ већ постоји, односно `true` у случају успешног уметања)
 - брисање кључа из стабла (враћа вредност `false` ако кључ не постоји, односно `true` у случају успешног брисања) уз логичко уклањање из базе свих филмова који одговарају задатом кључу
4. **[20 поена]** ефикасна претрага базе података која садржи податке о филмовима
 - проналажење свих филмова из задатог периода
 - одређивање статистике издавања филмова за изабране године

Обратити пажњу да се након логичког брисања записа о филму мора извршити ажурирање ретког индекса у одређеним ситуацијама.

Уколико се читају из датотеке, подаци о филмовима су задати у *Comma-Separated Values* (CSV) датотеци са заглављем следећег формата:

<code>original_language,original_title,release_date,revenue,runtime</code>
--

За раздвајање поља се користи знак зарез (comma ,).

Напомене

У задатку по потреби реализовати и додатне методе, где је то примерено. Написати главне програме који кориснику омогућавају рад са задатим стаблом, путем једноставног интерактивног менија.

Рад са датотекама у језику C++ захтева увожење заглавља `fstream` (именски простор `std`). За читање података користи се класа `ifstream`. Након отварања датотеке, читање се врши на исти начин као и са стандардног улаза. Кратак преглед најбитнијих метода и пријатељских функција ове класе је дат у наставку.

<pre>void open(const char * _Filename, ios_base::openmode _Mode = ios_base::in, int _Prot = (int)ios_base::_Openprot);</pre>	Отвара датотеку задатог имена за читање. <code>ifstream dat;</code> <code>dat.open("datoteka.txt");</code>
<pre>void close();</pre>	Затвара датотеку.
<pre>bool is_open();</pre>	Утврђује да ли је датотека отворена.
<pre>operator>></pre>	Преклопљен оператор за просте типове података.
<pre>ifstream dat; dat.open("datoteka.dat"); if(! dat.is_open()) greska(); char niz[20]; dat >> niz; dat.close();</pre>	Пример отварања датотеке, провере да ли је отварање успешно, читање једног знаковног низа из датотеке и затварања датотеке.