

ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2
2023-2024
- други домаћи задатак -

Опште напомене:

1. Домаћи задатак 2 састоји се од једног програмског проблема. Студенти проблеме решавају **самостално**, на програмском језику C или C++.
2. 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.
3. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Унос података треба омогућити било путем читања са стандардног улаза, било путем читања из датотеке.
5. Решења треба да буду отпорна на грешке и треба да кориснику пруже јасно обавештење у случају детекције грешке.
6. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као решење проблема које може освојити максималан број поена.**
7. 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). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Предаја домаћег задатка ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене.
9. Одбрана домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака, као и да пријаве теже случајеве повреде Правилника о дисциплинској одговорности студената Универзитета у Београду Дисциплинској комисији Факултета.

Задатак 1 – Ефикасно претраживање података из више табела базе података [100 поена]

Дате су две табеле *CUSTOMER* и *CUSTOMER_ACCOUNT* из једног стандардног *benchmark*-а за тестирање перформанси база података.

Структура табеле *CUSTOMER* је (C_ID, C_F_NAME, C_L_NAME, C_EMAIL_1, C_AD_ID). C_ID је примарни кључ табеле (целобројни податак). Остали атрибути су подаци везани за клијента (име, презиме, e-mail и адреса¹).

Цела табела је дата у текстуалној датотеци (*Customer.txt*) која има следећу структуру (| преставаља делимитер у датотеци):

C_ID|C_F_NAME|C_L_NAME|C_EMAIL_1|C_AD_ID

Табела *CUSTOMER_ACCOUNT* има структуру (CA_ID, CA_B_ID, CA_C_ID, CA_NAME, CA_TAX_ST, CA_BAL). CA_ID је примарни кључ табеле (целобројни податак). Параметар CA_C_ID (целобројног типа) представља страни кључ из табеле *CUSTOMER* (C_ID из табеле *CUSTOMER* одговара атрибуту CA_C_ID из табеле *CUSTOMER_ACCOUNT*). Параметри CA_B_ID, CA_TAX_ST су административни подаци везани за рачун (целобројног типа), CA_NAME представља име рачуна (string), а CA_BAL представља стање на рачуну (реалан број).

Напомена: Један корисник (*CUSTOMER*) може имати више рачуна (*CUSTOMER_ACCOUNT*).

Табела је дата у текстуалним датотекама (*CustomerAccount20.txt*, *CustomerAccount250.txt*, *CustomerAccount2500.txt*) које имају следећу структуру (| преставаља делимитер у датотеци):

CA_ID|CA_B_ID|CA_C_ID|CA_NAME|CA_TAX_ST|CA_BAL

Датотеке се разликују по броју редова, а дато их је више да би се уочио ефекат ефикасног претраживања са порастом величине табеле.

Потребно је имплементирати Б стабло реда m (где је m минимално 3, а максимално 10) као индексну структуру (у наставку индекс) над табелом *CUSTOMER_ACCOUNT*, погодну за брзу претрагу података о рачунима корисника.

Напомена: Конкретан избор кључа за претрагу у стаблу као и реализације дохватања информационог садржаја је предмет оцењивања и оставља се студенту.

Реализовати следеће операције над индексом:

1. **[20 поена]** Стварање и уништавање индекса над табелом *CUSTOMER_ACCOUNT*. Приликом стварања индекса корисник уноси ред Б стабла.
2. **[5 поена]** Испис индекса на стандардни излаз тако да су појединачни чворови уочљиви (међусобно размакнути). Ова ставка ће бити тестирана за мале величине табела (репрезентативна величина табеле за ову ставку је дата у датотеци *CustomerAccount20.txt*)
3. **[20 поена]** Исписивање података о кориснику и **ефикасна** претрага свих његових рачуна. Корисник уноси примарни кључ (C_ID). Подаци од кориснику се дохватају из текстуалне датотеке *Customer.txt* док је податке о рачунима потребно ефикасно дохватити из индекса. Претрага треба да на стандардни излаз испише укупан број корака за које је целокупна претрага завршена, као и да креира

¹ Атрибут C_AD_ID представља страни кључ из табеле која садржи податке о адресама. Детаљи те табеле нису од интереса за овај домаћи задатак.

датотеку са пронађеним записима, према формату (обратити пажњу да нису сви подаци из обе табеле потребни):

C_ID|C_F_NAME|C_L_NAME|C_EMAIL_1|CA_ID|CA_C_ID|CA_TAX_ST|CA_BAL

Уколико се деси да корисник не поседује ни један рачун, датотеку попунити једним редом у формату:

C_ID|C_F_NAME|C_L_NAME|C_EMAIL_1|null|null|null|null

4. **[20 поена]** Додавање новог записа (рекорда) у индекс (уносе се сви подаци о рачуну). Потребно је, на стандардан излаз, пријавити грешку уколико се примарни кључ понови. Сматрати да ће се уносити само рачуни за које постоји *CUSTOMER*, односно није потребно проверавати да ли постоји *CUSTOMER* задатог *CA_C_ID* у *Customer.txt* фајлу.
5. **[20 поена]** Брисање записа из индекса (уносе се сви подаци о рачуну који се брише). Потребно је, на стандардан излаз, исписати број корака за које је пронађен запис који се брише. Уколико није обрисан ни један запис, исписати информациону поруку о томе, уз број корака.
6. **[15 поена]** Брисање свих рачуна једног *CUSTOMER*-а. Задаје се параметар *CA_C_ID*. Потребно је, на стандардан излаз, исписати укупан број корака за које је целокупна операција завршена, као и све записе који су обрисани. Уколико није обрисан ни један запис, исписати информациону поруку о томе.

Корисник са програмом интерагује путем једноставног менија. Програм треба да испише садржај менија, а затим да чека да корисник изабере (унесе путем тастатуре) редни број неке од понуђених ставки, након чега, пре извршења, од корисника очекује да по потреби унесе додатне параметре. Поступак се понавља све док корисник у менију не изабере опцију за прекид програма.

Помоћ: рад са датотекама у језику C++

Рад са датотекама у језику C++ захтева увођење 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. <pre>ifstream dat; dat.open("datoteka.txt");</pre>
<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.