

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 2**2024-2025****- први домаћи задатак -****Опште напомене:**

1. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмском језику C или C++.
2. Пре одбране, сви студенти раде тест знања који се ради на рачунару коришћењем система *Moodle* (<http://elearning.rcub.bg.ac.rs/moodle/>). **Сви студенти треба да креирају налог и пријаве се на курс пре почетка лабораторијских вежби.** Пријава на курс ће бити прихваћена и важећа само уколико се студент региструје путем свог налога електронске поште на серверу **mail.student.etf.bg.ac.rs**.
3. Реализовани програм треба да комуницира 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 успешно 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 матрице и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
8. Одбрана првог домаћег задатка ће се обавити према распореду који ће накнадно бити објављен на сајту предмета.
9. За решавање задатака који имају више комбинација користити следеће формуле. (**R** – редни број индекса, **G** – последње две цифре године уписа):

$$i = (R + G) \bmod 2$$

$$j = (R + G) \bmod 3$$

10. Имена датотека која се предају морају бити **asp2dz1p1.c(pp)** и **asp2dz1p2.c(pp)**. Погрешно назван и предат домаћи задатак се кажњава са одузетих 25 поена.
11. Предаја домаћих ће бити омогућена преко *Moodle* система. Детаљније информације ће бити благовремено објављене.
12. 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. Током израде решења није дозвољена употреба алата вештачке интелигенције заснованих на великим језичким моделима (*ChatGPT*, *Github Copilot* и сл.).

Задатак 1 – имплементација именика помоћу стабла бинарног претраживања [50 поена]

У апликацији за комуникацију постоје записи о контактима корисника. Запис садржи барем презиме контакта (секвенцу карактера), име контакта (секвенцу карактера) и број телефона (секвенцу цифара). Ради брже претраге контакта из именика апликације за комуникацију, погодно је организовати именик са контактима у виду стабла бинарног претраживања. Прецизна структура записа који представља контакт се оставља студенту на имплементацију.

Потребно је конструисати стабло бинарног претраживања минималне висине на основу именика који је дат у виду текстуалне датотеке (*Contacts20.txt* и *Contacts100.txt*). Сваки ред именика садржи презиме контакта (секвенцу карактера), име контакта (секвенцу карактера) и број телефона (секвенцу цифара). Табела је сортирана лексикографски неоппадајуће по презимену, а потом, такође, лексикографски неоппадајуће по имену у случају више контакта са истим презименом. Дозвољено је да постоје два или више контакта са истим именом и презименом, али не и два контакта са истим бројем телефона.

Потребно је подржати следеће операције за рад са имеником у виду стабла бинарног претраживања:

- [10 поена] Учитавање именика и формирање стабла бинарног претраживања минималне висине,
- [10 поена] Ефикасна претрага свих контаката по презимену или презимену и имену,
- [10 поена] Измена презимена, имена или броја телефона постојећег контакта,
- [5 поена] Додавање новог контакта у именик,
- [5 поена] Уклањање контакта са датим именом и презименом из именика,
- [5 поена] Форматиран испис садржаја стабла,
- [5 поена] Брисање стабла из меморије.

У зависности од редног броја i проблема који се решава, додавање контакта са именом и презименом који већ постоје у именику реализовати на следећи начин:

0. Уметањем контакта као **следбеника** контакта са истим именом и презименом
1. Уметањем контакта као **претходника** контакта са истим именом и презименом

У зависности од редног броја j проблема који се решава, брисање стабла из меморије реализовати на следећи начин:

0. Користећи *preorder* обилазак стабла
1. Користећи *inorder* обилазак стабла
2. Користећи *level order* обилазак стабла

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.

Задатак 2 – одржавање линеарних структура података [50 поена]

На основу претходног решења из задатка 1 или на други погодан начин имплементираног именика, потребно је имплементирати додатне функционалности у раду апликације за комуникацију. Поред осталих, могуће су следеће функционалности:

- **[20 поена]** *Слање текстуалне поруке контакту.*
Претходни контакти са којима је корисник комуницирао налазе се у историји комуникације. Историја је ограничена на n контаката. Приликом слања нове поруке најпре се претражује историја комуникације, а, потом, ако је комуникација већ постојала, запис о њој се пребацује на почетак историје. У супротном, формира се нови запис о контакту и он се поставља на почетак историје комуникације. Записи могу и да нестану из историје, ако постоји бар n скорије тражених контаката. Параметар n задаје корисник.
- **[20 поена]** *Дељење садржаја контакту.*
Контакти којима је корисник претходно делио садржај (музичке записе, слике, итд.) сачувани су у историји дељења. Број контаката у историји дељења није ограничен. Приликом дељења садржаја најпре се претражује историја дељења, а, потом, ако је комуникација већ постојала, запис о њој се пребацује на једну позицију пре текуће (транспозиција). У супротном, формира се нови запис о контакту и он се поставља на крај историје дељења.

Историја комуникације и историја дељења могу бити реализоване као низ или листа записа. У зависности од редног броја i проблема и који се решава, реализовати ове две историје респективно као:

0. Низ и уланчану листу
1. Уланчану листу и низ

У зависности од редног броја j проблема који се решава, имплементирати једну од следећих хибридних техника одржавања историје дељења **[10 поена]**:

0. Уколико број дељења садржаја неком контакту порасте преко задате границе x , уместо за једну позицију, тај контакт се пребације на почетак историје дељења.
1. Уколико у току x узастопних дељења садржаја било којим контактима, садржај није подељен са неким контактом, запис о том контакту се помера на крај историје.
2. Ако се садржај наизменично дели са два корисника узастопце више од x пута, и транспозиција последично не доприноси побољшању перформанси, применити пребацивање на почетак.

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. Проширити претходни програм из првог дела задатка тако да укључи ставке реализоване у оквиру другог дела задатка и представља целовито решење заједно са њима.