ОДСЕК ЗА СОФТВЕРСКО ИНЖЕЊЕРСТВО

АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1

2020-2021

- први домаћи задатак -

Опште напомене:

1. 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**.
2. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблем решавају **самостално**, на програмском језику C или Python.
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 путем читања са стандардног улаза.
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.**
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 домаћег задатка ће се обавити у **понедељак, 22.03.2021. и уторак, 23.03.2021.** 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) mod 4

10. Предаја домаћих ће бити омогућена преко Moodle система до **понедељка, 22.03.2021. у 10:00**. Детаљније информације ће бити благовремено објављене на предметном сајту.
11. Имена датотека које се предају мора бити **dz1p1.(c|py)** и **dz1p2.(c|py)**
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

14.03.2021. године *Са предмета*

Задатак 1 – примена уланчаних листа [50 поена]

[**45 поена**] Написати интерактиван програм који коришћењем одговарајућег типа уланчаних листа илуструје рад са скуповима и полиномима.

Скуп се може имплементирати коришћењем растуће уређене уланчане листе. Један чвор листе представља један целобројни елемент скупа. За рад са скупом је потребно имплементирати следеће основне операције: проверу припадности елемента скупу, одређивање кардиналности скупа, додавање елемента у скуп, брисање елемента из скупа, унију, пресек и разлику два скупа. Приликом спровођења операције уније, пресека и разлике два скупа као резултат се формира нови скуп.

Полином са једном независном променљивом се може имплементирати коришћењем опадајуће уређене уланчане листе. Један чвор листе представља један члан полинома и садржи информације о коефицијенту и експоненту. Листа је уређена опадајуће по експоненту. У листи се чувају само чланови са ненултим коефицијентом. За рад са полиномом је потребно имплементирати следеће основне операције: додавање члана полинома, брисање члана полинома, рачунање вредности полинома за унету вредност независне променљиве, сабирање и множење два полинома. Приликом спровођења операција сабирања и множења два полинома као резултат се формира нови полином.

Зависно од редног броја проблема *i*, саставити **један** од следећих програма, који:

0. Приказује рад са **скупом** имплементираним коришћењем двоструко уланчане листе.
1. Приказује рад са **скупом** имплементираним коришћењем двоструко уланчане кружне листе.
2. Приказује рад са **полиномом** имплементираним коришћењем једноструко уланчане листе са заглављем.
3. Приказује рад са **полиномом** имплементираним коришћењем једноструко уланчане кружне листе са заглављем.

[**5 поена**] 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. За практичну примену, корисник програма треба да има најмање следеће могућности реализоване путем одговарајућих ставки менија:

1. учитавање новог скупа или полинома
2. додавање члана скупа или полинома
3. брисање члана скупа или полинома
4. испис задатог скупа или полинома
5. спровођење реализованих операција (више ставки)
6. брисање задатог скупа или полинома

Програм треба да има могућност манипулације са највише пет скупова или полинома истовремено. Корисник бира са којим скуповима или полиномима ради приликом задавања одговарајуће операције. Уколико нема места за смештање новог полинома, програм треба да обавести корисника да није могуће спровести акцију или да резултат смести на место једног од постојећих.

Задатак 2 – имплементација редова за чекање [50 поена]

[**45 поена**] Написати програм који симулира упис студената у наредну наставну годину. Сваког студента карактерише име, презиме, број индекса (јединствен идентификатор), студијски програм, тренутна година студија. Потребно је имплементирати симулацију уписа школске године на шалтеру студентске службе на следећи начин:

0. По покретању програма учитавају се подаци о студентима са стандардног улаза и смештају се у уланчану листу. По завршетку се приказује порука на стандардном излазу о тренутном броју студената у листи.
1. Врши се симулација уписа. Креира се псеудослучајан цео број који мора бити мањи од тренутног броја студената у листи и студент са дате позиције се смешта у нову структуру података која симулира ред чекања испред студентске службе. Након премештања студента у ред чекања, студент се брише из листе. Овај поступак се понавља све док листа не остане празна.
2. Из реда чекања студенти „долазе“ на шалтер студентске службе. За сваког студента се генерише псеудослучајан број између 0 и 1 који представља вероватноћу да ће упис бити обављен успешно. Уколико је генерисани број већи од задатог прага X ($0 \leq X \leq 0.5$), студент обавља упис и уклања се из реда по завршетку. На стандардном излазу се исписује име и презиме студента и коју годину је уписао. У супротном, студент се пребацује на крај реда.
3. Поступак уписа се понавља док ред за чекање не остане празан. Када је ред празан, симулација се завршава и на стандардном излазу се приказује информација о броју корака симулације.

Ред за чекање имплементирати истом врсте структуре података која је дефинисана у Задатку 1. Потребно је најмање имплементирати следеће операције: дохватање елемента из реда, стављање елемента у ред, проверу да ли је ред празан и проверу да ли је ред пун.

[**5 поена**] 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.

Додатак – имплементација уланчане листе на различитим програмским језицима

У прилогу се налазе примери дефиниције чвора и заглавља једноструко уланчане листе целих бројева на различитим програмским језицима. Дате дефиниције се могу користити као узор за реализацију студентских решења.

Програмски језик C

```
typedef struct listNode {
     int info;
     struct listNode *next;
} ListNode;
typedef struct listHeader {
     ListNode *head, *tail;
     int numElem;
} ListHeader;
```

Програмски језик Python

```
class ListNode:
    def __init__(self, info=None):
        self.info = info
        self.next = None
class ListHeader:
    def __init__(self):
        self.head = None
        self.tail = None
        self.numElem = None
```