

**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1**  
**2016-2017**  
**- домаћи задатак 1 -**

**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**. Приступна лозинка за курс ће бити послата путем одговарајуће листе електронске поште за предмет.

2. Домаћи задатак 1 састоји се од два програмска проблема. Студенти проблеме решавају **самостално**, на програмским језицима Pascal или C, по избору.
3. Реализовани програми треба да комуницирају са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
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.
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 матрице и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
7. Одбрана првог домаћег задатка ће се обавити у **понедељак, 13.03.2017. и уторак, 14.03.2017.** према распореду који ће накнадно бити објављен на сајту предмета.
8. Формула за редни број проблема **i** који треба решавати у **првом** и **другом** задатку је следећа (R – редни број индекса, G – последње две цифре године уписа):  
први задатак:             $i = (R + G) \bmod 4 + 1$   
други задатак:         $i = (R + G) \bmod 3 + 1$
9. Имена датотека које се предају морају бити **dz1p1.(pas|c)** и **dz1p2.(pas|c)**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

06.03.2017. године

*Са предмета*

## Задатак 1 – имплементација стека (50 поена)

Написати програм на програмском језику Pascal или C који илуструје рад са стеком. Елементи стека су цели бројеви. Програм треба омогући стварање стека (са задатим максималним капацитетом у случају секвенцијалне имплементације), као и следеће основне операције: уметање елемента на стек, уклањање елемента са стека (уз враћање вредности), дохватање елемента са врха стека (без уклањања), дохватање броја заузетих места на стеку, дохватање максималног капацитета стека. Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената стек над којим се врше операције. 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.

Зависно од редног броја проблема, написати **једну** од следећих имплементација стека:

1. Секвенцијална имплементација, показивач врха стека показује на прву слободну локацију.
2. Секвенцијална имплементација, показивач врха стека показује на последњу заузету локацију.
3. Уланчана имплементација (једноструко уланчана листа).
4. Уланчана имплементација (једноструко уланчана кружна листа).

Саму структуру података моделовати помоћу кориснички дефинисаног типа (записа) који ће поред линеарне структуре података (низа или листе) употребљене за реализацију стека садржати и додатне податке о самој структури података (попут тренутне и максималне величине, показивача врха стека и сл.).

## Задатак 2 – алгоритми за рад са ретко поседнутим матрицама (50 поена)

Написати интерактиван програм на програмском језику Pascal или C који илуструје рад са ретко поседнутим матрицама. Ретко поседнуте матрице су врста матрица код којих већина елемената има подразумевану вредност. Код оваквих матрица се могу направити значајне уштеде у простору потребном за њихово складиштење уколико се експлицитно памте само вредности које нису подразумеване. У наставку је дат кратак опис четири технике смештања ретко поседнутих матрица.

**Вектор записа од три поља (coordinate liste – COO)** је начин смештања ретко поседнуте матрице димензија  $M \times N$  код кога се подразумеване вредности памте експлицитним чувањем тројки које чине врсту, колону и вредност елемента. Тројке се у вектору смештају сортиране најпре по редном броју врсте, а затим по редном броју колоне.

**Compressed row storage (CSR)** формат чува ретко поседнуту матрицу димензија  $M \times N$  помоћу три независна вектора ( $V, C, R$ ). Вектор  $V$  садржи вредности и има онолико елемената колико има подразумеваних елемената матрице. Вектор  $C$  садржи број колоне одговарајућег елемента вектора  $V$ . Сваки елемент вектора  $R$  одговара једној врсти матрице и садржи индекс првог елемента вектора  $C$  и  $V$  који одговара тој врсти. Вектор  $R$  има дужину  $M+1$ , а у последњем елементу се чува укупан број подразумеваних елемената матрице.

**Compressed sparse column (CSC)** формат чува ретко поседнуту матрицу димензија  $M \times N$  на сличан начин као CSR формат, осим што вектори  $R$  и  $C$  замењују места и значење. Вектор  $V$  садржи вредности и има онолико елемената колико има подразумеваних елемената матрице. Вектор  $R$  садржи број врсте одговарајућег елемента вектора  $V$ . Сваки елемент вектора  $C$  одговара једној колони матрице и садржи индекс првог елемента вектора  $R$  и  $V$  који одговара тој колони. Вектор  $C$  има дужину  $N+1$ , а у последњем елементу се чува укупан број подразумеваних елемената матрице.

Зависно од редног броја проблема, саставити **један** од следећих програма, који представља ретко поседнуту матрицу помоћу:

1. једног вектора записа од три поља
2. три независна вектора у CSR формату
3. три независна вектора у CSC формату

Програм треба да омогући стварање матрице и унос ретко поседнуте матрице задатих димензија и задатог броја ненултих елемената, дохватање вредности елемента задате врсте и колоне, испис матрице (укључујући и елементе подразумеване вредности). Није неопходно омогућити накнадну измену унете матрице. Посебно, зависно од редног броја проблема, саставити **један** од следећих потпрограма који омогућавају следећу операцију за рад са ретко поседнутом матрицом:

1. множење две матрице уз проверу да ли се могу помножити
2. сабирање две матрице уз проверу да ли се могу сабрати
3. разлаже матрицу на збир једне горње троугаоне и једне строго доње троугаону матрицу уз проверу да ли се може разложити

**Напомена:** у матрици је потребно чувати само неподразумеване елементе, а резултат операције треба да има исти формат као полазне матрице. Више информација о наведеним техникама за смештање ретко поседнутих матрица се може добити на следећој адреси: [https://en.wikipedia.org/wiki/Sparse\\_matrix](https://en.wikipedia.org/wiki/Sparse_matrix).