

**АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА 1**  
**2016-2017**  
**- домаћи задатак 2 -**

**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, уколико то већ нису учинили. 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 2 састоји се од једног програмског проблема. Студенти проблем решавају **самостално**, на програмском језику Pascal или C, по избору.
3. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
4. Решење треба да буде отпорно на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
5. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Коришћење рекурзије у решењима није дозвољено, као ни готових структура података.**
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 **петак, 21.04.2017.** према распореду који ће благовремено бити објављен на сајту предмета.
8. Формула за редни број комбинације проблема **i** и **j** који треба решавати у задатку је следећа: (R – редни број индекса, G – последње две цифре године уписа):
$$i = (R + G) \bmod 3 + 1$$
$$j = (R + G) \bmod 4 + 1$$
9. Име датотеке која се предаје мора бити **dz2p1.(pas|c)**
10. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

## Задатак – игра погађања појмова (100 поена)

Написати програм који имплементира модификацију познате игре погађања појмова „20 Questions guessing game“. Игра се одвија тако што корисник замисли један појам, а рачунар покушава да га погоди на основу постојеће базе знања. Рачунар поставља питање по питање, на која корисник одговара искључиво са „Да“ или „Не“, а на основу тих одговора рачунар покушава да одгонетне појам који је замислио. Ако рачунар успешно погоди појам који је корисник замислио, он побеђује. Уколико појам не постоји у бази знања, база се проширује, а корисник побеђује.

За потребе овог задатка, у оквиру игре могу постојати три основне категорије: животиње, биљке и предмети. Уколико рачунар не погоди појам, корисник треба да му сугерише питање на основу кога ће разликовати појам који је рачунар понудио као тачан и појам који је корисник замислио.

Један типичан сценарио игре следи. Корисник на почетку игре замисли животињу, на пример кокошку. Програм ће му на основу претходно унесене базе знања постављати питања. Прво питање би било „Да ли је у питању животиња?“, како би одредила припадност одговарајућој категорији, на које корисник одговара са „Да“. Следеће питање би било: „Да ли је у питању сисар?“, на које корисник одговара са „Не“. Затим би могло следити питање: „Да ли животиња има крила?“, а корисник би одговорио „Да“. Наредно питање би могло бити: „Да ли животиња лети?“, корисник би одговорио са „Не“. Коначно, након што би исцрпео сва питања у бази, програм би покушао да погоди појам: „Да ли је животиња кокошка?“, а корисник би одговорио са „Да“, чиме би рачунар победио у игри. Уколико у бази знања не постоји појам „Кокошка“, већ „Ној“, рачунар би као последње питање поставио „Да ли је животиња ној?“, а корисник би одговорио са „Не“, чиме би остварио победу. У том случају, рачунар би затражио да корисник допуни базу знања. Најпре би га упутио да унесе појам (кокошка), а затим би затражио да унесе питање на основу кога се може разликовати појам који је он замислио од појма који је као одговор понудио рачунар: „Унесите питање којим могу да разликујем појам ној од појма кокошка“, чиме би било извршено ажурирање базе знања.

Базу знања у оквиру игре моделовати коришћењем **бинарног стабла одлучивања**. Унутрашњи чворови стабла ће представљати питања, а листови појмове. За сваку категорију у оквиру базе треба формирати засебно стабло, тако да базу знања чини шума одговарајућих стабала за сваку категорију. Потребно је омогућити кориснику следеће операције:

- a) [40 поена] Формирање базе знања уношењем питања и одговора са стандардног улаза
- b) [10 поена] Брисање базе знања
- c) [25 поена] Симулацију игре

Зависно од дефинисаног редног броја **i** имплементирати и једну од следећих операција:

1. [15 поена] Испис свих питања у бази знања
2. [15 поена] Испис свих одговора у бази знања
3. [15 поена] Одређивање висине стабала у бази знања

Зависно од дефинисаног редног броја **j** потребне обиласке у стаблу имплементирати **искључиво** коришћењем:

1. *preorder* алгоритма
2. *inorder* алгоритма
3. *postorder* алгоритма
4. *level order* алгоритма

Приликом формирања базе знања, евентуално претходно формирана база знања се брише. Сматрати да ниједно питање није дуже од 255 знакова. Након завршетка симулације игре, корисник може поново да игра или да се врати у главни мени.

**[10 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.

Више информација о оригиналној игри се може видети: [https://en.wikipedia.org/wiki/Twenty\\_Questions](https://en.wikipedia.org/wiki/Twenty_Questions). Веб верзија игре се може играти на <http://www.20q.net/>.