

ОДСЕК ЗА РАЧУНАРСКУ ТЕХНИКУ И ИНФОРМАТИКУ
АЛГОРИТМИ И СТРУКТУРЕ ПОДАТАКА
2018-2019

- домаћи задатак -

Опште напомене:

1. Домаћи задатак састоји се од два програмска проблема. Студенти проблем решавају **самостално**, на програмском језику C++. **Није дозвољено коришћење готових структура података из STL и сличних библиотека.**
2. Право да раде домаћи задатак имају сви студенти који прате предмет. Студенти треба да се пријаве за одбрану домаћег задатка најкасније до **понедељка, 24.12.2018.** године путем одговарајућег линка на сајту предмета.
3. Домаћи задатак замењује други део испита, односно задатак на испиту који се пише у конкретном језику. Поени са домаћег задатка ће стога бити скалирани у одговарајућем односу. Одбраном домаћег задатка студент се одриче права да ради други део испита. Студент на испиту треба да назначи да ли је радио домаћи задатак како би му били признати поени.
4. Реализовани програм треба да комуницира са корисником путем једноставног менија који приказује реализоване операције и омогућава сукцесивну примену операција у произвољном редоследу.
5. Унос података треба омогућити било путем читања са стандардног улаза.
6. Решења треба да буду отпорна на грешке и треба да кориснику пружи јасно обавештење у случају детекције грешке.
7. Приликом оцењивања, биће узето у обзир рационално коришћење ресурса. **Примена рекурзије се неће признати као успешно решење проблема и неће бити оцењена са максималним бројем поена.**
8. 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 и слично). Неоправдано увођење ограничавајуће претпоставке повлачи негативне поене.
9. Одбрана домаћег задатка ће се обавити у **петак, 28.12.2018.** према распореду који ће накнадно бити објављен на сајту предмета.
10. Формула за редни број комбинације проблема **i** и **j** који треба решавати је следећа:
(R – редни број индекса, G – последње две цифре године уписа):

$$i = (R + G) \bmod 3 + 1$$

$$j = (R + G) \bmod 2 + 1$$

11. Имена датотеке које се предаје мора бити **dzp1.(c|cpp)** и **dzp2.(c|cpp)**
12. Предметни наставници задржавају право да изврше проверу сличности предатих домаћих задатака и коригују освојени број поена након одбране домаћих задатака.

Задатак 1 – Имплементација стабла [50 поена]

[30 поена] Написати програм који илуструје рад са стаблом целих бројева реда m ($m \geq 2$). Програм треба да омогући стварање стабла задатог степена, уметање новог елемента, испис стабла, као и брисање стабла. Уметање новог елемента извршити тако да стабло увек буде комплетно или скоро комплетно. Чвор стабла садржи и показивач на родитељски чвор.

[15 поена] Зависно од дефинисаног редног броја i имплементирати и једну од следећих операција:

1. Одређивање висине стабла
2. Одређивање ширине стабла
3. Одређивање броја чворова стабла

Све наведене операције треба реализовати путем одговарајућих потпрограма чији је један од аргумената стабло над којим се врше операције.

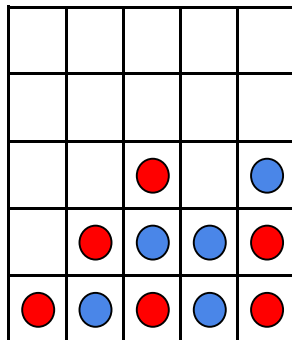
Зависно од редног броја j потребне обиласке у стаблу имплементирати искључиво коришћењем:

1. *preorder* алгоритма
2. *level order* алгоритма

[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.

Задатак 2 – Имплементација игре [50 поена]

Модификовањем претходно написаног решења, написати програм за реализацију једне варијанте познате игре са решетком димензија 5x5 у коју се убацују куглице. Игру играју два играча којима је на почетку игре додељена боја (плава или црвена). Играчи повлаче потезе наизменично тако што у сваком потезу убацују по једну куглицу боје која им је додељена на почетку у неку од колона решетке. Куглица упада у колону на сам врх претходно убачених куглица у истој колони. Победник је онај играч који први повеже 3 куглице исте боје. Куглице могу бити повезане хоризонтално, вертикално или дијагонално. Циљ сваког потеза неког играча је да за себе створи најповољнију позицију, а да свог противника осујети у његовој стратегији. На слици је приказано једно победничко стање за играча са црвеном бојом.



Програму се задаје тренутно стање игре (уносом матрице која представља решетку). Програм од задатог почетног стања треба да формира стабло игре уз оцене сваког стања (1 - потез који првог играча води до победе, 0 - нерешено, -1 - позиција која другог играча води до победе). На почетку игре се на случајан начин додељује боја на основу чега се одређује ко први игра. Сваки чвор стабла представља стање игре након неког потеза – распоред куглица у решетки у одговарајућем потезу. Корену стабла одговара задато стање решетке, док се могуће крајње позиције налазе у листовима стабла. Сматрати да је унето почетно стање игре валидно.

[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 барем следеће могућности реализоване путем одговарајућих ставки менија:

1. **[30 поена]** Започињање нове игре и формирање стабла
2. **[15 поена]** Испис стабла игре

Потребне обиласке у стаблу имплементирати искључиво коришћењем алгоритма за обилазак који је дефинисан у првом делу задатка.

Више информација о проблему се може видети на следећим страницама:

https://en.wikipedia.org/wiki/Connect_Four

<https://www.coolmathgames.com/0-4-in-a-row>