# 2014/2015

## A

1. doplnit kód Queup a Remove (přednáška 5, strana 27)  
2. převedení příkladu z infixu do postfixu (přednáška 5, strana 6)  
3. popsat pseudopříkaz vlastními slovy, max 30 slov (strana 8: [https://wis.fit.vutbr.cz/FIT/st/course- ... f?cid=9958](https://wis.fit.vutbr.cz/FIT/st/course-files-st.php/course/IAL-IT/lectures/Opora-IAL-2014-verze-14-N.pdf?cid=9958) )  
4. průchod binárním stromem, napsat PreOrder, InOrder, PostOrder a InvPreOrder (přednáška 5, strana 62, více pomůže tento odkaz <http://en.wikipedia.org/wiki/Tree_traversal>, u invertovaných cestách se postupuje v opačném směru šipky)

## B

1. průchod binárním stromem, napsat PreOrder, InOrder, PostOrder a InvPreOrder  
---I---  
--/----  
A-----  
--\----  
--- H--  
--/----  
B------  
--\----  
--- G—

2. Převedení infixového zápisu na postfix  
(A-B)/(C+D)\*(E-F)+(G+H)\*I/J =

3. Definice příkazu  
  
4. Sekvenční vyhledávání v poli Tab s adaptivním uspořádáním podle četnosti vyhledávání se zarážkou.  
Počet prvků pole N, pro výměnu prvků použíte ":=:" . Všechny prázdné kolonky **nemusí** být použity!  
*(\*píšu tento kód z hlavy, takže očekávejte nějaké chybky...\*)  
(\*používám pomlčky na odsazení řádků\*)*

**begin  
 i:=1;  
 <vynechany řádek>  
 while K<>T.Tab[i].Klic do i:=i+1;  
 <vynechany řádek>  
 if i<>(N+1)   
 if i<>1   
 <vynechany řádek>  
 end  
 Search := true;  
 end  
 <vynechaný řádek>  
end;**

# 2013/2014

## A

1. Doplnit kod na operace QueUp a Remove   
2. Diagram signatury ADT fronta  
3. Napsat rekurzivní definici pro ekvivalenci dvou stromu.  
4. Dijkstrova varianta binarniho vyhledavani - doplnit kod (*podminky u klicu, napsat na kolikatem prvku se nachazi vysledek)*

## B

1. Doplnit kód na operace nejlev a order   
2. Diagram signatury ADT jednosměrný seznam  
3. Napsat rekurzivní definici binarního stromu.  
4. Doplnit kód - Sekvenční vyhledávání v poli se zarážkou

# 2012/2013

## A

1. Zadany strom, vypsat ho systemem PreOrder, InOrder, PostOrder (3x 1 bod)  
2. Vypocitat vyraz v postfixove a prefixove notaci (2x 1 bod)  
3. Namalovat diagram signatury ADT zasobnik (1x 3 body)  
4. Doplnit kod do nejake Pascalovske hruzy. Domnivam se, ze to byla procedura pro vyrazeni prvku z obousmerne vazaneho seznamu pracujici s indexy misto ukazatelu *(tzv. Honzik-uzivatelska-dynamicka-pamet)*, ale Buh sud, co tim myslel doopravdy *(2x 2 body -- ale byla tam tri volna mista :roll:)*  
5. Rekurzivni definice delky seznamu, do 25 slov (1x 2 body)

## B

1.Priechody stromom(In,Pre,Postorder)   
2.Vycislenie infix/postfix  
3.Diagram signatury - Tabulka  
4.dopisanie kodu - deleteDMA  
5.ekvivalencia zoznamu - rekurzivna definicia

# 2011/2012

## A

1. Doplnenie funkcie **deleteDMA** *(4b)*  
2. Doplnenie funkcii **Queue (Full, Remove, QueUp)** *(5b)*  
3. Diagram signatury ADT zoznam *(3b)*  
4. Definovat dlzku zoznamu rekurzivne na max. 25 slov. *(2b)*

## B

1. Doplneni funkce deleteDMA (4b)  
2. Doplneni funkci Queue (Full, Remove, QueUp) - zřetězeným seznamem (5b)  
3. Diagram signatury ADT fronta (3b)  
4. Definovat ekvivalenci seznamu rekurzivne na max. 25 slov. (2b)