

1. Na vstupu jsou dána celá čísla uspořádaná od nejmenšího k největšímu. Napište program v Pascalu, který co nejefektivněji ze zadaných čísel vytvoří vyvážený *binární vyhledávací strom*. U vyváženého stromu se délka nejkratší a nejdelší větve liší nejvýše o 1. Analyzujte počet kroků algoritmu. Využijte následující datové typy reprezentující strom:

```

type Uk = ^Uzel;
      Uzel = record
              data: integer;    { uložená informace }
              levy, pravy: Uk     { levý a pravý syn }
      end;

```

(4 body)

2. Napište proceduru v Pascalu, která obrátí pořadí prvků v daném *jednosměrném lineárním spojovém seznamu* celých čísel. To jest například ze seznamu obsahujícího po řadě hodnoty 2, 8, 4, 9 chceme vytvořit seznam s prvky v pořadí 9, 4, 8, 2. Analyzujte počet kroků algoritmu. Využijte následující datové typy reprezentující seznam:

```

type Uk = ^Uzel;
      Uzel = record
              data: integer;    { uložená informace }
              dalsi: Uk          { následník }
      end;

```

(4 body)

3. Napište proceduru v Pascalu, která korektně zruší zadaný *jednosměrný lineární spojový seznam*. To znamená, že chceme uvolnit veškerou paměť, kterou seznam obsazuje. Využijte datové typy z úlohy 2.

(4 body)

4. Napište proceduru v Pascalu, která korektně zruší zadaný *binární vyhledávací strom*. To znamená, že chceme uvolnit veškerou paměť, kterou strom obsazuje. Využijte datové typy z úlohy 1.

(4 body)

5. (**prémiová**) Napište proceduru v Pascalu, která zruší všechny prvky se zadanou hodnotou v zadaném *binárním vyhledávacím stromu*. Předpokládejte, že strom může obsahovat více uzlů se stejnou hodnotou. Analyzujte počet kroků algoritmu. Využijte datové typy z úlohy 1.

(8 bodů)