

*Zadania oddajemy w osobnych plikach czytelnie podpisane i skomentowane. Każde rozwiązanie należy uzasadnić, i podać koszty czasowy i pamięciowy w zależności od liczby węzłów  $n$  oraz wysokości  $h$  drzew, a w zadaniu 1 też od  $k$ . Wolno korzystać z własnych notatek. Nie wolno kontaktować się z kimkolwiek i używać internetu do innych celów, niż komunikacja z moodlem.*

### Zadanie E1

Proszę napisać funkcję

`bool istnieje(Twezel* d, int k),`

która przekaże w wyniku wartość `true` wtedy i tylko wtedy gdy istnieje wartość  $x$  taka, że wartości  $x$  oraz  $x - k$  występują w drzewie binarnych wyszukiwań  $d$ .

### Zadanie E2

Napisz funkcję

`int rowneH(Twezel* d),`

która wyznaczy liczbę nieuporządkowanych par różnych poddrzew [drzewa](#)  $d$  o identycznej wysokości. [Drzewa](#) są różne, jeśli adresy ich korzeni są różne.



### Zadanie E3

Powiemy, że drzewo binarne  $d$  o  $n$  węzłach jest *intrygujące*, jeśli można ponumerować jego węzły liczbami od 1 do  $n$  w taki sposób, że:

- każdy węzeł będący lewym synem ma numer nieparzysty;
- każdy węzeł będący prawym synem ma numer parzysty;
- przy takim numerowaniu drzewo  $d$  jest BST względem takiej numeracji, czyli w każdym węźle jego numer jest większy od wszystkich numerów w lewym poddrzewie i mniejszy od wszystkich numerów w prawym poddrzewie.

Korzeń  $d$  może mieć wartość o dowolnej parzystości. Napisz funkcję

`bool Intrygujące (Twezel *d),`

która stwierdzi, czy drzewo  $d$  jest intrygujące i, jeśli jest, to w węzłach wstawi taką intrygującą numerację, a jeśli nie jest, to pozostawi drzewo z niezmienionymi wartościami.