

Trær – Case 4:

(Teksten er hentet fra oppg.nr.3 på eksamen 19/12-2017 i IMT2021 - Algoritmske metoder)

NB: *Alle* deloppgavene er *totalt* uavhengige. Men, alle bygger på binære trær bestående av:

```
struct Node {  
    char ID;           //  Nodens ID/key/nøkkel/navn (en bokstav).  
    Node *left, *right; //  Peker til venstre/høyre subtre (evt. nullptr).  
    Node(const char id) { ID = id;  left = right = nullptr; }  
};
```

Vi har også den globale variabelen:

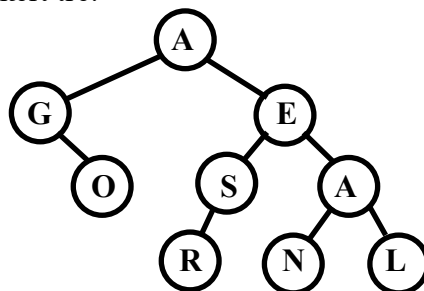
```
Node* gRoot = nullptr; //  Rot-peker til treet.
```

Vi forutsetter nå videre at det ifm. hver deloppgave er bygd opp ulike binære trær som gRoot peker til.

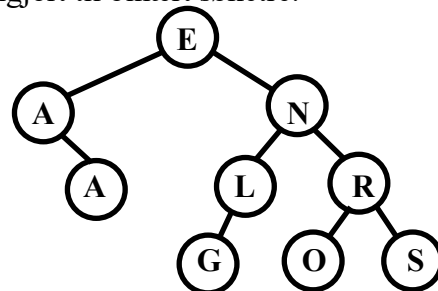
a) Skriv koden som trengs for å finne nivået for bladen som ligger på det høyeste oddetalls nivået. Rota ligger på nivå nr.0. Er det bare bladen på partalls nivå, eller at treet er tomt, skal dette max. nivået settes til -1 (minus en).

b) Skriv koden som trengs for å konvertere/omgjøre et binært tre (max 100 noder) til et binært søketre, uten å endre treets originale struktur/form. F.eks:

Initielt binært tre:



Omgjort til binært søketre:



NB: For oppgave a og b (men *ikke* oppgave c) gjelder: Kan du gjenbruke kode fra eksempler (EKS_xx.CPP) er det bare å henvise til dette. Trenger du flere globale *enkelt-variable* og/eller *en hjelpearray*, er det også lov å innføre dette.

c) Vi innfører de tre globale pekerne (og *kun* disse): Node *gNode1, *gNode2, *gForrige; Vi har et binært søketre, der *to* (og *kun to*) av noderes ID har byttet plass (dvs. treet blir ordnet/sortert igjen om disse to bare bytter ID).

Lag den rekursive funksjonen void finnNoder(Node* node)
som finner og setter gNode1 og gNode2 til å peke på de aktuelle nodene.

Kode (som du *ikke* skal skrive) vil etterpå bytte om disse to noderes ID.

NB: Vi forutsetter at treet inneholder minst tre noder, og at de to nodene *ikke* kommer rett etter hverandre i inorder rekkefølge.